

ISO TC184/SC4/WG3 N799
1999-09-30

This document is a pre-Committee Draft (pre-CD) version of ISO 10303-218, and is intended for review within ISO TC184/SC4/WG3/T23, *Ships* team. Readers should contact their national standards bodies or the ISO TC184/SC4 Secretariat (sc4sec@cme.nist.gov) for the status of the actual Committee Draft (CD) document.

ISO/CD 10303-218

Product data representation and exchange: Application Protocol: Ship Structures

COPYRIGHT NOTICE: This ISO document is a working draft or committee draft and is copyright protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by Participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purposes of selling it should be addressed as shown below (*via* the ISO TC184/SC4 Secretariat's member body) or to ISO's member body in the country of the requester.

Copyright Manager
ANSI
11 West 42nd Street
New York, New York 10036
USA
phone: +1-212-642-4900
fax: +1-212-398-0023

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement. Violators may be prosecuted.

ABSTRACT:

This document defines an Application Protocol for the product data pertaining to the pre-design, main design and production engineering of the ship's structural systems.

KEYWORDS: application protocol,ships,ship structures

COMMENTS TO READER:

This document contains the Scope statement, Units of Functionality, Application objects and assertions, Application Intertreted, Reference and Activity Model, and mapping tables for the Ship Structures AP. This version of AP218 reflects the result of interpretation workshops carried out during 1998-1999. This document is prepared for AP218 CD ballot.

Project Leader: Matthias Grau
Address: KCS Consulting GmbH
Bramfelder Str. 121
22305 Hamburg
Germany

Telephone: +49 (40) 61148-173
Telefacsimile: +49 (40) 61148-191
Electronic mail: mg@kcs.com

Project Editor: Yuanxie Janke-Zhao
Address: KCS Consulting GmbH
Bramfelder Str. 121
22305 Hamburg
Germany

Telephone: +49 (40) 61148-154
Telefacsimile: +49 (40) 61148-191
Electronic mail: yjz@kcs.com

Contents	Page
1 Scope	1
2 Normative references	4
3 Terms, definitions and abbreviations	5
3.1 Terms defined in ISO 10303-1	5
3.2 Terms defined in ISO 10303-21	7
3.3 Terms defined in ISO 10303-31	7
3.4 Terms defined in ISO 10303-41	7
3.5 Terms defined in ISO 10303-42	8
3.6 Terms defined in ISO 10303-43	9
3.7 Other definitions	9
3.8 Abbreviations	20
4 Information requirements	22
4.1 Units of functionality	22
4.1.1 class_approvals	23
4.1.2 configuration_management	24
4.1.3 definitions	25
4.1.4 design_loads	26
4.1.5 external_references	27
4.1.6 hull_cross_sections	28
4.1.7 integrated_resources	28
4.1.8 items	28
4.1.9 location_concept	29
4.1.10 product_structures	30
4.1.11 representations	31
4.1.12 shapes	31
4.1.13 ship_general_characteristics	32
4.1.14 ship_manufacturing_definitions	33
4.1.15 ship_materials	34
4.1.16 ship_measures	34
4.1.17 structural_features	37
4.1.18 structural_parts	39
4.1.19 structural_systems	41
4.1.20 support_resources	42
4.1.21 welds	43
4.2 Application objects	45
4.3 Application assertions	279
5 Application interpreted model	308
5.1 Mapping table	308
5.2 AIM EXPRESS short listing	713
6 Conformance requirements	1064

Annex A (normative)	AIM EXPRESS expanded listing	1066
Annex B (normative)	AIM short names	1067
Annex C (normative)	Implementation method specific requirements	1068
Annex D (normative)	Protocol Implementation Conformance Statement (PICS) proforma	1069
Annex E (normative)	Information object registration	1070
E.1	Document identification	1070
E.2	Schema identification	1071
Annex F (informative)	Application activity model	1072
F.1	Application activity model definitions and abbreviations	1072
F.2	Application activity model diagrams	1097
Annex G (informative)	Application reference model	1123
Annex H (informative)	AIM EXPRESS-G	1191
Annex J (informative)	AIM EXPRESS listing	1192
Annex K (informative)	Application protocol usage guide	1193
Annex L (informative)	Technical discussions	1194
L.1	Introduction	1194
L.2	The Ship Common Model	1196
L.3	Key concepts of ship structures	1199
Bibliography	1210
Index	1212

Figures

1	Shipbuilding application protocols	xv
2	Data planning model	xvii
3	An Angle_bar_cross_section	46
4	An Assembly	49
5	Bedding moment	54
6	Bevel_corner_cutout	55
7	A Bulbflat_cross_section	59
8	One side butt weld	61
9	Both sides butt weld	61
10	A hull cross section of a bulk carrier	147
11	A doublebottom of a bulk carrier	148
12	Circular_cutout	148
13	Corrugated system parameters	149
14	A corrugated bulkhead	149

15	Corrugated part parameters	150
16	Design_still_water_bending_moment	150
17	Drain_hole_cutout	151
18	Elliptical_cutout	151
19	Elongated_oval_cutout	152
20	A Flat_bar_cross_section	152
21	Global axis placements	153
22	An example of groove weld	153
23	A hull cross section of a cargo ship	154
24	Inward_round_corner_cutout and Outward_round_corner_cutout	155
25	Rectangular_cutback_corner_cutout	182
26	Round_corner_rectangular_cutout	185
27	Shear_bevel_corner_cutout	189
28	Spacing positions	195
29	Spot and plug weld	198
30	Plate cutouts	203
31	Profile cutouts	204
32	A deck	207
33	A bulkhead	208
34	Structural_system	217
35	A doublebottom as section	218
36	A doublebottom	219
37	A DECK	225
38	A single bottom	226
39	A double bottom	227
40	A SHELL	267
41	A 2-dimensional FORE SHIP	268
42	The FORE SHIP in a 3-dimensional view	268
43	A bow	269
44	A TANK	270
45	A KEEL	271
46	A WEB FRAME	272
47	A LONGITUDINAL GIRDER	273
48	A T_bar_cross_section	274
49	Examples of welds	274
50	Weld positions	275
51	Weld methods	275
52	Backstep and wandering methods of welding	276
53	Weld procedures	276
54	Weld types	277
55	Weld sequences	278
F.1	IDEF0 basic notation	1097
F.2	Node A-0: Ship Life Cycle Description	1098
F.3	Node A0: Perform Ship Life Cycle	1099
F.4	Node A1: Specify Ship	1100
F.5	Node A12: Prepare Bid	1101

F.6	Node A122: Create Preliminary Design	1102
F.7	Node A2: Complete and Approve Ship Design	1103
F.8	Node A24: Complete Design of Ship Structure	1104
F.9	Node A241: Design Ship Structure	1105
F.10	Node A2412: Create Design of Classification Items	1106
F.11	Node A24121: Design of Transverse Structure	1107
F.12	Node A242: Approve Design of Ship Structure	1108
F.13	Node A2424: Check Global Strength and Secondary Structure	1109
F.14	Node A243: Elaborate Ship Structure Design	1110
F.15	Node A2434: Make Hull Detail Design	1111
F.16	Node A24341: Design Secondary Structure	1112
F.17	Node A24342: Handle Plates	1113
F.18	Node A24343: Handle Profiles	1114
F.19	Node A244: Make Production Design of Ship Structure	1115
F.20	Node A2442: Create Production Documents	1116
F.21	Node A2443: Refine Working Information	1117
F.22	Node A3: Produce and Inspect a Ship	1118
F.23	Node A34: Test a Ship	1119
F.24	Node A4: Operate and Maintain a Ship	1120
F.25	Node Hierarchy of the Application Activity Model	1121
F.26	Node Hierarchy of the Application Activity Model (continued)	1122
G.1	class_approvals_structure schema (1/3) in the class_approvals UoF	1124
G.2	class_approvals_structure schema (2/3) in the class_approvals UoF	1125
G.3	class_approvals_structure schema (3/3) in the class_approvals UoF	1126
G.4	approvals schema (1/1) in the configuration_management UoF	1127
G.5	changes schema (1/2) in the configuration_management UoF	1128
G.6	changes schema (2/2) in the configuration_management UoF	1129
G.7	events schema (1/1) in the configuration_management UoF	1130
G.8	versions schema (1/1) in the configuration_management UoF	1131
G.9	definitions schema (1/1) in the definitions UoF	1132
G.10	design_loads schema (1/2) in the design_loads UoF	1133
G.11	design_loads schema (2/2) in the design_loads UoF	1134
G.12	documents schema (1/1) in the external_references UoF	1135
G.13	external_references schema (1/1) in the external_references UoF	1136
G.14	library_references schema (1/2) in the external_references UoF	1137
G.15	library_references schema (2/2) in the external_references UoF	1138
G.16	hull_cross_section_design_definitions schema (1/1) in the hull_cross_sections UoF	1139
G.17	hull_cross_sections schema (1/1) in the hull_cross_sections UoF	1140
G.18	generic_product_structures schema (1/1) in the items UoF	1141
G.19	ships schema (1/1) in the items UoF	1142
G.20	global_axis_characteristics schema (1/1) in the location_concept UoF	1143
G.21	local_co_ordinate_systems schema (1/1) in the location_concept UoF	1144
G.22	local_co_ordinate_systems_with_station_reference schema (1/1) in the location_concept UoF	1145
G.23	spacing_grids schema (1/1) in the location_concept UoF	1146
G.24	feature_design_definitions schema (1/1) in the product_structures UoF	1147

G.25	features schema (1/1) in the product_structures UoF	1148
G.26	parts schema (1/1) in the product_structures UoF	1149
G.27	product_structure_by_assembly schema (1/1) in the product_structures UoF	1150
G.28	product_structure_by_system schema (1/1) in the product_structures UoF	1151
G.29	designation_characteristics schema (1/2) in the ship_general_characteristics UoF . .	1152
G.30	designation_characteristics schema (2/2) in the ship_general_characteristics UoF . .	1153
G.31	dimension_characteristics schema (1/1) in the ship_general_characteristics UoF . .	1154
G.32	freeboard_characteristics schema (1/1) in the ship_general_characteristics UoF . .	1155
G.33	lightship_weight schema (1/1) in the ship_general_characteristics UoF	1156
G.34	ship_types schema (1/1) in the product_definitions UoF	1157
G.35	weight_data schema (1/1) in the ship_general_characteristics UoF	1158
G.36	assembly_manufacturing_definitions schema (1/1) in the ship_manufacturing_defin- itions UoF	1159
G.37	structural_part_manufacturing_definitions schema (1/1) in the ship_manufacturing_ definitions UoF	1160
G.38	structural_part_manufacturing_features_desig_definitions schema (1/1) in the ship_ manufacturing_definitions UoF	1161
G.39	structural_part_manufacturing_features schema (1/1) in the ship_manufacturing_ definitions UoF	1162
G.40	weld_manufacturing_definitions schema (1/1) in the ship_manufacturing_definitions UoF	1163
G.41	ship_material_properties schema (1/1) in the ship_materials UoF	1164
G.42	measures schema (1/4) in the ship_measures UoF	1165
G.43	measures schema (2/4) in the ship_measures UoF	1166
G.44	measures schema (3/4) in the ship_measures UoF	1167
G.45	measures schema (4/4) in the ship_measures UoF	1168
G.46	corner_cutout_design_definitions schema (1/1) in the structural_features UoF	1169
G.47	corner_cutouts schema (1/1) in the structural_features UoF	1170
G.48	edge_cutout_design_definitions schema (1/1) in the structural_features UoF	1171
G.49	edge_cutouts schema (1/1) in the structural_features UoF	1172
G.50	interior_cutout_design_definitions schema (1/2) in the structural_features UoF . . .	1173
G.51	interior_cutout_design_definitions schema (2/2) in the structural_features UoF . . .	1174
G.52	interior_cutouts schema (1/1) in the structural_features UoF	1175
G.53	position_feature_design_definitions schema (1/1) in the structural_features UoF . .	1176
G.54	position_features schema (1/1) in the structural_features UoF	1177
G.55	structural_features schema (1/1) in the structural_features UoF	1178
G.56	structural_part_design_definitions schema (1/4) in the structural_parts UoF	1179
G.57	structural_part_design_definitions schema (2/4) in the structural_parts UoF	1180
G.58	structural_part_design_definitions schema (3/4) in the structural_parts UoF	1181
G.59	structural_part_design_definitions schema (4/4) in the structural_parts UoF	1182
G.60	structural_parts schema (1/1) in the structural_parts UoF	1183
G.61	structural_system_design_definitions schema (1/2) in the structural_systems UoF .	1184
G.62	structural_system_design_definitions schema (2/2) in the structural_systems UoF .	1185
G.63	structural_systems schema (1/2) in the structural_systems UoF	1186
G.64	structural_systems schema (2/2) in the structural_systems UoF	1187
G.65	welded_joints schema (1/1) in the welds UoF	1188

G.66	welds schema (1/2) in the welds UoF	1189
G.67	welds schema (2/2) in the welds UoF	1190
L.1	Ship product model	1194
L.2	Life cycle concept of ship's structures	1196
L.3	SCM framework	1197
L.4	Subtypes of Items for ship structures	1200
L.5	Subtypes of Definition for ship structures	1201
L.6	Subtypes of Design_definition for ship structures	1201
L.7	Corrugated part model	1203
L.8	Corrugated system model	1204
L.9	Hierarchy of feature design definitions	1205

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303-218 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration – Product data representation and exchange*:

- Part 1, Overview and fundamental principles;
- Part 11, Description methods: The EXPRESS language reference manual;
- Part 12, Description method: The EXPRESS-I language reference manual;
- Part 21, Implementation methods: Clear text encoding of the exchange structure;
- Part 22, Implementation method: Standard data access interface specification;
- Part 23, Implementation method: C++ language binding to the standard data access interface;
- Part 24, Implementation method: C language binding to the standard data access interface;
- Part 26, Implementation method: Interface definition language binding to the standard data access interface;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 34, Conformance testing methodology and framework: Abstract test methods;
- Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;

- Part 41, Integrated generic resources: Fundamentals of product description and support;
- Part 42, Integrated generic resources: Geometric and topological representation;
- Part 43, Integrated generic resources: Representation structures;
- Part 44, Integrated generic resources: Product structure configuration;
- Part 45, Integrated generic resource: Materials;
- Part 46, Integrated generic resources: Visual presentation;
- Part 47, Integrated generic resource: Shape variation tolerances;
- Part 49, Integrated generic resource: Process structure and properties;
- Part 101, Integrated application resource: Draughting;
- Part 104, Integrated application resource: Finite element analysis;
- Part 105, Integrated application resource: Kinematics;
- Part 106, Integrated application resource: Building construction core model;
- Part 107, Engineering Analysis Core Application reference model (EA C-ARM);
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 204, Application protocol: Mechanical design using boundary representation;
- Part 205, Application protocol: Mechanical design using surface representation;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 208, Application protocol: Life cycle management — Change process;
- Part 209, Application protocol: Composite and metallic structural analysis and related design;
- Part 210, Application protocol: Electronic assembly, interconnect, and packaging design;
- Part 212, Application protocol: Electrotechnical design and installation;

- Part 213, Application protocol: Numerical control process plans for machined parts;
- Part 214, Application protocol: Core data for automotive mechanical design processes;
- Part 215, Application protocol: Ship arrangement;
- Part 216, Application protocol: Ship moulded forms;
- Part 217, Application protocol: Ship piping;
- Part 218, Application protocol: Ship structures;
- Part 221, Application protocol: Functional data and their schematic representation for process plant;
- Part 222, Application protocol: Exchange of product data for composite structures;
- Part 223, Application protocol: Exchange of design and manufacturing product information for casting parts;
- Part 224, Application protocol: Mechanical product definition for process plans using machining features;
- Part 225, Application protocol: Building elements using explicit shape representation;
- Part 226, Application protocol: Ship mechanical systems;
- Part 227, Application protocol: Plant spatial configuration;
- Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;
- Part 230, Application protocol: Building structural frame: Steelwork;
- Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;
- Part 232, Application Protocol: Technical data packaging core information and exchange;
- Part 301, Abstract test suite: Explicit draughting;
- Part 302, Abstract test suite: Associative draughting;
- Part 303, Abstract test suite: Configuration controlled design;
- Part 304, Abstract test suite: Mechanical design using boundary representation;

- Part 305, Abstract test suite: Mechanical design using surface representation;
- Part 307, Abstract test suite: Sheet metal die planning and design;
- Part 308, Abstract test suite: Life cycle management — Change process;
- Part 309, Abstract test suite: Composite and metallic structural analysis and related design;
- Part 310, Abstract test suite: Electronic assembly, interconnect, and packaging design;
- Part 312, Abstract test suite: Electrotechnical design and installation;
- Part 313, Abstract test suite: Numerical control process plans for machined parts;
- Part 314, Abstract test suite: Core data for automotive mechanical design processes;
- Part 315, Abstract test suite: Ship arrangement;
- Part 316, Abstract test suite: Ship moulded forms;
- Part 317, Abstract test suite: Ship piping;
- Part 318, Abstract test suite: Ship structures;
- Part 321, Abstract test suite: Functional data and their schematic representation for process plant;
- Part 322, Abstract test suite: Exchange of product data for composite structures;
- Part 323, Abstract test suite: Exchange of design and manufacturing product information for casting parts;
- Part 324, Abstract test suite: Mechanical product definition for process plans using machining features;
- Part 325, Abstract test suite: Building elements using explicit shape representation;
- Part 326, Abstract test suite: Ship mechanical systems;
- Part 327, Abstract test suite: Plant spatial configuration;
- Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;
- Part 330, Abstract test suite: Building structural frame: Steelwork;
- Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;

- Part 332, Abstract test suite: Technical data packaging core information and exchange;
- Part 501, Application interpreted construct: Edge-based wireframe;
- Part 502, Application interpreted construct: Shell-based wireframe;
- Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;
- Part 504, Application interpreted construct: Draughting annotation;
- Part 505, Application interpreted construct: Drawing structure and administration;
- Part 506, Application interpreted construct: Draughting elements;
- Part 507, Application interpreted construct: Geometrically bounded surface;
- Part 508, Application interpreted construct: Non-manifold surface;
- Part 509, Application interpreted construct: Manifold surface;
- Part 510, Application interpreted construct: Geometrically bounded wireframe;
- Part 511, Application interpreted construct: Topologically bounded surface;
- Part 512, Application interpreted construct: Faceted boundary representation;
- Part 513, Application interpreted construct: Elementary boundary representation;
- Part 514, Application interpreted construct: Advanced boundary representation;
- Part 515, Application interpreted construct: Constructive solid geometry;
- Part 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded representation.
- Part 519, Application interpreted construct: Geometric tolerances;
- Part 520, Application interpreted construct: Associative draughting.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Parts 11 to 12 specify the description methods,
- Parts 21 to 26 specify the implementation methods,

- Parts 31 to 35 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,
- Parts 101 to 106 specify the integrated application resources,
- Parts 201 to 232 specify the application protocols,
- Parts 301 to 332 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

Annexes A, B, C, D and E are an integral part of this part of ISO 10303. Annexes F, G, H, J, K and L are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the application protocols series.

This part of ISO 10303 specifies an application protocol (AP) for the definition of ship structures and related data.

This application protocol defines the context, scope and information requirements for the communication of ship structural data and specifies the use of the integrated resources necessary to satisfy these requirements.

The series of shipbuilding industry application protocols assumes that the ship product model can be divided into separate ship systems that each covers a key element of the ship for its whole life cycle. These key elements are: Major subdivisions of this International Standard are:

- ship arrangements;
- ship moulded forms
- ship distribution systems (piping, heating, ventilation and air conditioning, as well as electrical and cableway);
- ship structures;
- ship mechanical systems (cargo handling, machinery and propulsion);
- ship outfit and furnishings;
- combat systems;
- communication;
- navigation;
- operation.

Each separate system is described by one or more different application protocols. The full series of shipbuilding application protocols is depicted in figure 1.

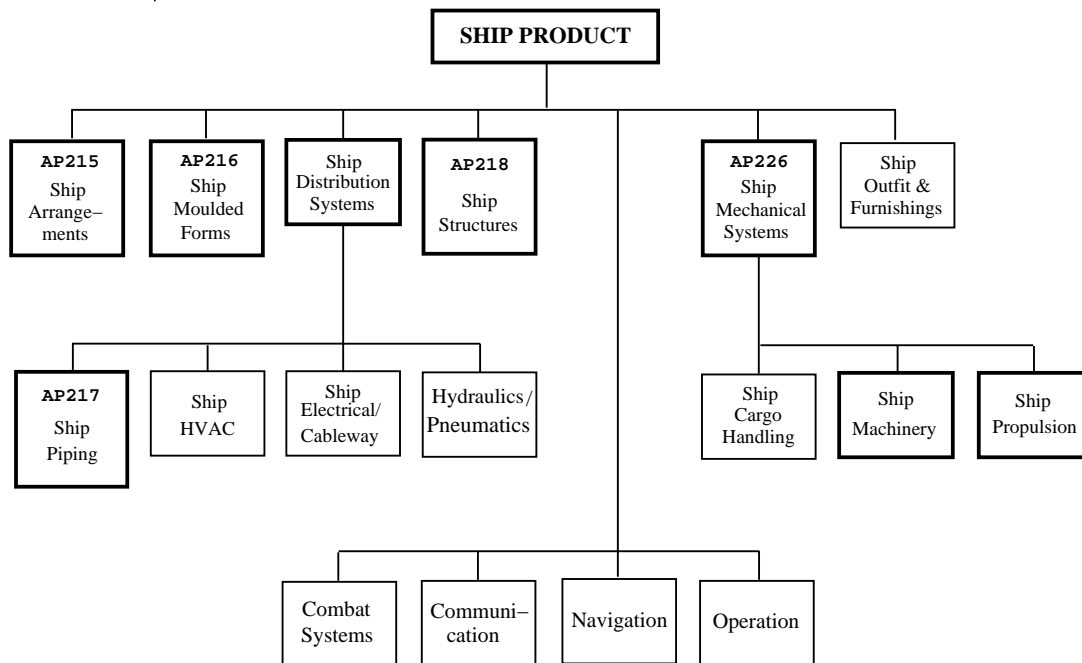


Figure 1 – Shipbuilding application protocols

Those aspects of the ship product model that are common to each ship application protocol are described consistently and identically in each application protocol.

Within the series of shipping application protocols this part of ISO 10303 details the key element of ship structures.

This part of ISO 10303 specifies an application protocol for the exchange of product data representing a ship's structures information between different organizations with a need for that data. Such organizations include ship owners, ship classification societies, design agencies and fabricators. This AP has been developed to support the shipbuilding activities and computer applications associated with the pre-design, the main design, the manufacturing, the maintenance (repair) and the inspection and survey during operation life cycle phases for commercial or naval ships.

This application protocol defines the context, scope and information requirements for the exchange of ship structures data and specifies the integrated resources necessary to satisfy these requirements.

Figure 2 illustrates how the ship structures information may be grouped under the following:

Major subdivisions of this International Standard are:

- ship common model;
- ship common utilities;
- design of ship structures;
- manufacturing of ship structures;
- class approvals;
- shape representations.

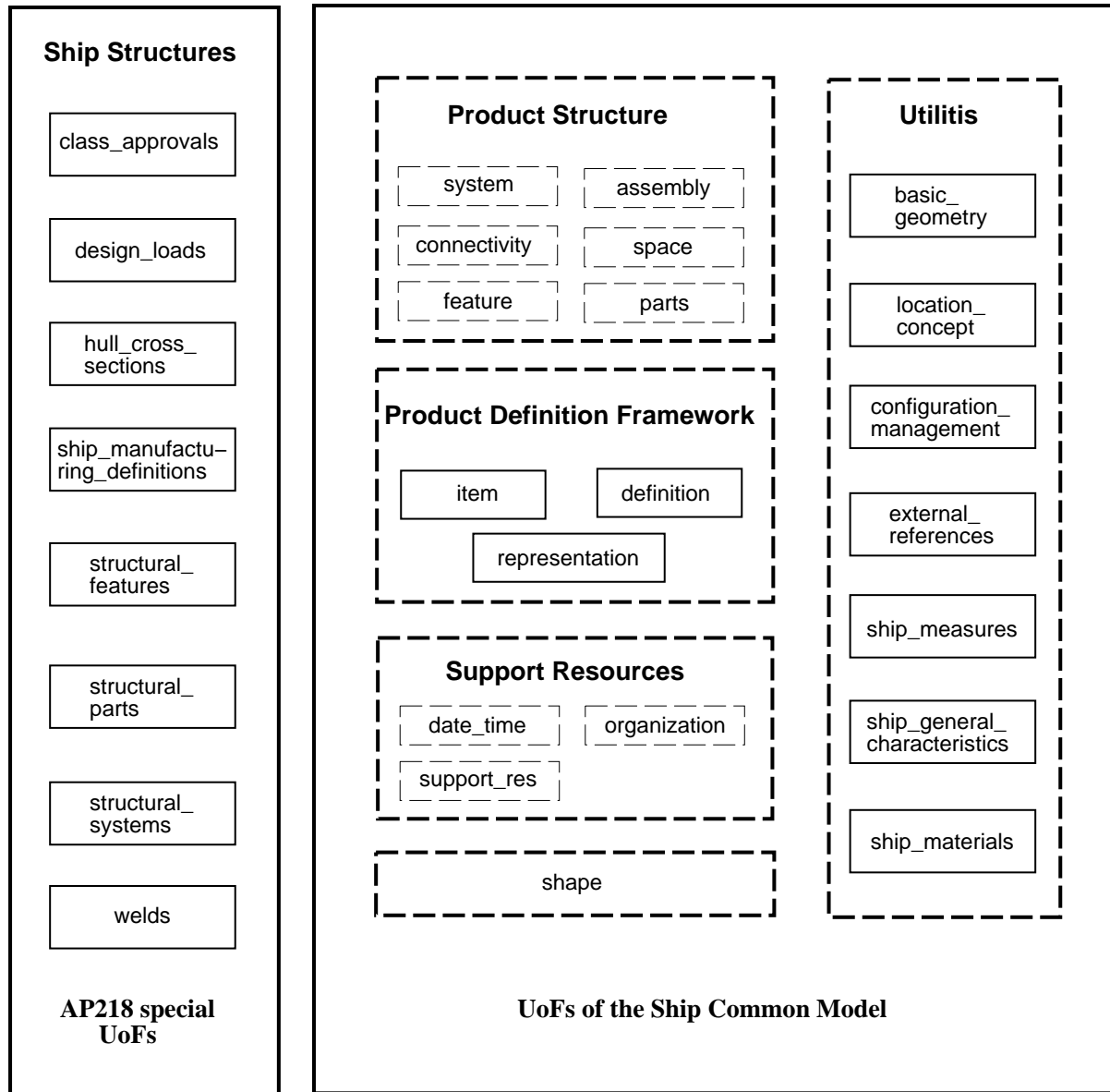


Figure 2 – Data planning model

The group ship common model specifies the ship building common model which is used for all ship building APs (currently for AP215, AP216, AP217 and AP218). This group is considered to present the framework of Shipbuilding APs. The group ship common model is basis for the design and manufacturing of ship structures.

The group ship common utilities includes the configuration management, the ship general characteristics, the external references, the location concept, the ship measures and the ship materials, they are all applied to the product ship structures as well as to the ship common model.

The group design of ship structures describes the design definition of ship structures during the predesign and design phase of a ship. This group includes design loads, hull cross sections, and the design definitions of structural parts, structural systems and structural features.

The group manufacturing of ship structures describes the manufacturing definitions of ship structures during the manufacturing phase of a ship. It includes manufacturing definitions for ship's structural parts and assemblies and the welding definitions for the welds used for the steel structural parts and assemblies.

The class approvals approves the structural parts and hull cross sections of the ship, with aspects of the design and manufacturing.

The group shape representations describes the geometrical and topological wireframes, the advanced B-rep and the topological surfaces that can be used to represent the shape of ship structures.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in annex F. The information requirements of the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in 5.1, shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in annex A contains the complete EXPRESS for the AIM without annotation. A graphical representation of the AIM is given in annex H. Additional requirements for specific implementation methods are given in annex D.

Industrial automation systems and integration — Product data representation and exchange — Part 218 : Application Protocol: Ship Structures

1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for the exchange of product definition data and its configuration and approval status information for ship structural systems. Configuration in this context pertains to data specific to revision tracking and change history of selected ship structural entities within the product model. Approval pertains to the company-internal approval and the classification society approval. This Application Protocol supports the shipbuilding activities and applications associated with the design phase and the manufacturing phase.

NOTE The application activity model in annex F provides a graphical representation of the processes and information flows which are the basis for the definition of the scope of this part of ISO 10303.

The following are within the scope of this part of ISO 10303:

- product definition data pertaining to the ship's structure which includes hull structure, superstructure and all other internal structures of commercial and naval ships;
- product definition data pertaining to the ship's structure, at the following phases of the ship's life cycle: Major subdivisions of this International Standard are:
 - a) pre-design phase;
 - b) the main design phase;
 - c) the manufacturing phase;
- product definition data pertaining to the product structuring of ships, including the structuring by system and by assemblies within the ship;
- product definition data identifying the ship's general characteristics which are relevant to the design of the ship's structure;

NOTE The general characteristics include ship's main dimensions, designations and principle characteristics, as well as the rules, regulations and standards applicable to the ship. It also includes lightships weight distribution and free-board characteristics for the purpose of design and design approvals.

- product definition data pertaining to the ship's global co-ordinate system, local co-ordinate systems and spacing grids, which are used for defining the geometry of the ship's structure;
- product definition data pertaining to the geometrical representation of the ship's structure parts and assemblies;
- product definition data pertaining to the hull plating and the stiffener profiles, and the definition of structural features, i.e., the edge, corner and interior cut-outs, which comprise the ship's structure parts and assemblies, including functional descriptions;
- product definition data pertaining to the design of the welded connections and joints of ship's structure parts and assemblies, including edge preparations and weld type and size;
- product definition data pertaining to the specification of transverse cross-sections through the ship's structure for the purpose of approval of strength;
- product definition data pertaining to ship's design loads, including shear forces and bending moments acting on the ship's structure, for the purpose of determining the longitudinal strength of the ship;
- product definition data pertaining to the weights and centres of gravity of the ship's structure parts and assemblies;
- product definition data pertaining to the materials of ship's structure, required to manufacture the ship or a part of it;
- product definition data pertaining to the configuration management of the ship's structure, including approval, versioning and change administration;
- product definition data pertaining to external references, technical documentation and other supporting concepts which are necessary and pertinent to the design and manufacture of the ship's structure parts and assemblies.

The following are outside the scope of this part of ISO 10303:

- product definition data pertaining to the ship's structure at the operation and de-commissioning phases of the ship's life cycle;
- business data for the management of a ship development project, such as budgets, schedules and resource requirements;
- product definition data pertaining to the direct calculation of ship's structure in the design stage;
- product definition data pertaining to the coating of structural parts as well as the production tolerances;

- product definition data pertaining to the ship's subdivision;
- product definition data pertaining to the ship's distribution systems;

EXAMPLE 1 The electrical, piping and HVAC systems

- product definition data pertaining to the ship's machinery and propulsion systems;
- product definition data pertaining to the ship's outfit and furnishing;
- product definition data pertaining to ship's hull structure parts which are manufactured by forging or casting.

EXAMPLE 2 Stern frames, rudder horns and propeller shaft brackets

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 31:1994, *Quantities and Units*.

ISO 690:1995, *Documentation - Bibliographic references - content, form, and Structure*.

ISO 1000:1992, *SI units and recommendations for the use of their multiples and of certain other units*.

ISO/IEC 8824-1:1994, *Information Technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1) - Part 1: Specification of Basic Notation*.

ISO/IEC Directives, Part 3 (Third edition, 1997), *Rules for the structure and drafting of International Standards*.

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles*.

ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*.

ISO 10303-21:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*.

ISO 10303-31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts*.

ISO 10303-41:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support*.

ISO 10303-42:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological Representation*.

ISO 10303-43:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resources: Representation structures*.

ISO 10303-44:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resources: Product Structure Configuration.*

ISO 10303-45, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resources: Materials.*

ISO 10303-47, *Industrial automation systems and integration — Product data representation and exchange — Part 47: Integrated generic resources: Shape variation tolerances.*

ISO/WD 10303-215, *Industrial automation systems and integration — Product data representation and exchange — Part 215: Application Protocol: Ship Arrangements.*

ISO/WD 10303-216, *Industrial automation systems and integration — Product data representation and exchange — Part 216: Application Protocol: Ship Moulded Forms.*

ISO/CD 10303-501, *Industrial automation systems and integration — Product data representation and exchange — Part 501: Application interpreted construct: Edge-based wireframe.*

ISO/CD 10303-508, *Industrial automation systems and integration — Product data representation and exchange — Part 508: Application interpreted construct: Non-manifold surface.*

ISO/CD 10303-510, *Industrial automation systems and integration — Product data representation and exchange — Part 510: Application interpreted construct: Geometrically bounded wireframe.*

ISO/CD 10303-511, *Industrial automation systems and integration — Product data representation and exchange — Part 511: Application interpreted construct: Topologically bounded surface.*

ISO/CD 10303-514, *Industrial automation systems and integration — Product data representation and exchange — Part 514: Application interpreted construct: Advanced boundary representation.*

The Shorter Oxford English Dictionary.,

The Concise Oxford English Dictionary.,

3 Terms, definitions and abbreviations

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

- abstract test suite (ATS);
- application;

- application activity model (AAM);
- application context;
- application interpreted model (AIM);
- application object;
- application protocol (AP);
- application reference model (ARM);
- computer aided design (CAD);
- computer aided manufacture (CAM);
- conformance class;
- conformance requirement;
- data;
- data exchange;
- generic resource;
- information;
- integrated generic resource;
- integrated resource;
- interpretation;
- model;
- PICS proforma;
- product;
- product data;
- product information model;
- resource construct;
- test purpose;

- unit of functionality (UoF);
- validation.

3.2 Terms defined in ISO 10303-21

- implementation method.

3.3 Terms defined in ISO 10303-31

- conformance testing;
- postprocessor;
- preprocessor.

3.4 Terms defined in ISO 10303-41

- address;
- approval_status;
- area_measure;
- date_and_time;
- derived_unit;
- identifier;
- label;
- length_measure;
- mass_measure;
- named_unit;
- organization;
- person;
- person_and_organization;
- plane_angle_measure;

- positive_length_measure;
- ratio_measure;
- shape_representation;
- text;
- time_measure;
- unit;
- volume_measure.

3.5 Terms defined in ISO 10303-42

- axis2_placement_3d;
- boundary;
- bounded_curve;
- box_domain;
- cartesian_point;
- circle;
- closed curve;
- connected;
- curve;
- edge;
- edge_curve;
- geometric_representation_context;
- line;
- open curve;
- orientable;
- placement;

- point;
- surface;
- vector;
- vertex.

3.6 Terms defined in ISO 10303-43

- representation;
- representation_context;
- representation_item;
- representation_relationship.

3.7 Other definitions

For the purposes of this part of ISO 10303, the following definitions apply:

3.7.1

aft perpendicular

a vertical line located at the intersection of the summer load (design) waterline and the after side of the rudder post or sternpost, or the centerline of the rudderstock if there is no rudder post or sternpost

3.7.2

amidships

the vertical plane arranged transversely at mid-length between the after perpendicular and the forward perpendicular

3.7.3

baseline

imaginary horizontal line used in design and production from which measurements are referenced (baseline is measure zero vertically)

3.7.4

beam

one type of structural stiffener which may or may not lie on a surface and may or may not be a profile

NOTE In a different context, beam is also the term used for one of the main dimensions.

3.7.5

bill of material (BOM)

a list of all raw material stock required to fabricate the product

NOTE The BOM may be for an individual part, for an aggregation of parts (i.e., an assembly), or for a selected volumetric region (i.e., a zone) of the ship.

3.7.6

builder

the company that built the ship

NOTE The builder may be further categorized as either the leadyard builder (that has constructed the first ship in a series or class) or a followyard builder.

3.7.7

builder hull number

the unique serial number assigned to the ship by the shipbuilder indicating where this ship stands with respect to all the hulls built by this yard

NOTE The first hull would be 1, the tenth would be 10, etc. This hull number also serves as a reference to the design information generated during the design and construction processes.

3.7.8

bulkhead

one type of hullform element representing a division of the internal space of a ship, equivalent to the function of a wall in a building

NOTE In case of a longitudinal bulkhead it may extend over one, several or all decks. Likewise it may extend over the complete length of a ship or only between certain transverse bulkheads. In case of a transverse bulkhead it may extend over completely from the bottom to the top of the ship, or only between certain decks. Likewise it may extend over the complete width of the ship or only between certain longitudinal bulkheads. Bulkheads are usually classified by their tightness

3.7.9

buttock

one type of hullform element representing the intersection of a vertical, longitudinal plane with the hull surface

3.7.10

camber

the change in vertical measurement of a deck relative to the deck at centerline

NOTE A positive value of a camber would mean that the deck at side is lower than the deck at centerline. Sometimes this value is expressed as a slope (such as X mm per Y m) or as an angle.

3.7.11

centerline

imaginary horizontal datum line used in design and production from which measurements are referenced

NOTE It divides the ship longitudinally into port and starboard sides.

3.7.12

centroid

the center of an item, area or volume measured with respect to some defined location

NOTE Within shipbuilding it is expressed as a vertical component measured above the baseline, a transverse component measured from the centerline, and a longitudinal component measured from either amidships or the forward Perpendicular.

3.7.13

chine

the intersection of the bottom and the sides of a flat or V-bottomed ship

3.7.14

classification

the process of ensuring that a ship is designed, built and maintained to a prescribed standard

NOTE This is done by periodic surveys of the ship.

3.7.15

combat systems

systems on board naval ships used for Communication and Battle

NOTE Communication is carried out for detection of targets, disturbing enemy communication signals and with weapon systems on board the ship. Battle systems are the weapon systems on board the ship.

3.7.16

compartments

one type of space that represents the spatial partitioning of the interior of the ship for the purposes of segregating dry and liquid cargo, passengers, crew, machinery, equipment, etc.

NOTE A compartment is represented by a closed, bounded volume defined in one of three ways:

- topologically by a collection of hullform elements (such as longitudinal and transverse bulkheads, decks, hull shell, etc.) and/or non structural joiner bulkheads;
- geometrically by one or more surfaces;
- by a combination of both.

3.7.17

configuration management

addresses the specific needs of the shipbuilding community with regards to the version control, approval status, product structure, and "as-built" condition of ship structure product model data

3.7.18

cutouts

the results of removing a portion of a structural part, usually along one of its edges, to allow a stiffener to pass through the part

NOTE Typically, these cutouts have standard patterns and are parametrically defined based on the type and size of the penetrating stiffener.

3.7.19

deadrise

one of two ways of defining the linear change in the vertical measurement of the bottom of the ship from some inboard point to some outboard point, traditionally from the keel to the turn of bilge

NOTE A X deadrise would mean that the bottom of the ship would be X mm higher at the outboard location than at the centre line. It is sometimes expressed as a slope (such as X mm per Y m).

3.7.20

deadweight

the difference between the lightship weight and loaded displacement

EXAMPLE 1 The weight of cargo plus weights of fuel, stores, water ballast, fresh water, crew and passengers, and baggage.

3.7.21

deck

one type of hullform element representing a vertical division of the internal space of a ship, equivalent to the function of a floor in a multi-storied building

NOTE It may extend over completely or only partially across the ship. Likewise it may extend over the complete length of the ship or only over selected portions of the length. It may be planar and orthogonal to the baseline (in which case its shape may be implicitly defined by a single vertical measurement), completely flat (i.e., planar) and not aligned with the baseline, or non-planar (i.e., 3-dimensional). Decks are usually designated by their location as boat deck, bridge, deck, etc. or by functional purpose such as freeboard deck and subdivision deck.

3.7.22

definition geometry

a designation for geometry that defines the shape representation of an item

NOTE There is a contractual implication with respect to guaranteed accuracy associated with this type of geometry.

3.7.23

depth

the vertical distance between the ship's baseline and a horizontal level in the ship

EXAMPLE 1 The depth to a specific waterline, depth to the main deck, depth to the uppermost continuous deck

3.7.24

design change process

the processing of a formal request for change to one or more configuration items during the design phase

3.7.25

design load

the load which the ship is designed to carry or to withstand

NOTE It is used to for instance present the design displacement, design strength.

3.7.26

design phase

stage within the ship life-cycle when the ship is being designed

NOTE A subdivision can be made into preliminary design, detailed design and production design.

3.7.27

design zone

an abstract boundary identifying a region of a ship with unique requirements or characteristics which must be specially treated in the design phase

NOTE These zones carry designations such as Fire Zone, CPS Zone, Subsafe Zone and Ship Work Authorization Boundary Zone.

3.7.28

diagonal

one type of hullform element representing the intersection of the hull surface with longitudinally oriented plane, inclined to both the vertical and transverse global co-ordinate system axes

3.7.29

engineering parts

pipng parts, HVAC parts, discrete parts, outfit and furnishing parts

3.7.30

explicit shape

the collection of geometry, conforming to specific geometric modelling representations such as CSG or B-rep, defining the shape of a structural part

NOTE There may be more than one explicit shape for a single part. The explicit shape may have embedded in it the geometry associated with the part's features.

3.7.31

flare

the transverse measurement indicating how much greater a waterline's half-breadth is compared to the half-breadth of the design waterline, measured at the same longitudinal reference

NOTE An X mm flare for a waterline means that the moulded hullform is $2xX$ mm wider at that waterline than it is at the design waterline.

3.7.32

forward perpendicular

a vertical line at the intersection of the foreside of the stem profile and the summer load (i.e., design waterline)

NOTE It is commonly used as a reference point for measuring or locating items longitudinally on a ship.

3.7.33

frame

one of the transverse members that make up the riblike part of the skeleton of a ship

NOTE The frames act as stiffeners, holding the outside plating in shape and maintaining the transverse form of the ship.

3.7.34

functional structural element

a classification of ship structure associated with a particular system view of the ship

3.7.35

geometry of the ship hull

the collection of geometry-elements that define the shape of the watertight envelope of the ship

NOTE This typically includes the underwater shell, the uppermost watertight deck or decks, and the internal watertight transverse and longitudinal bulkheads.

3.7.36

hogging

the situation of a vessel in still water as well as in a seaway, in which a vessel is bending: the deck of the ship would be in tension, the bottom will be in compression

3.7.37

hull number

a number assigned to the ship for its life. When combined with a naval ship type abbreviation, it identifies the vessel

3.7.38

hull shell

one type of hullform element representing the outermost, watertight envelope of a hullform section

NOTE Depending on the context of hullform section, this shell may represent the outer boundary of the ship itself, the rudder, sonar dome, etc.

3.7.39

hullform sections

entities indicating whether or not the hullform section has been faired

NOTE If the section is faired, then all hullform elements contained in the section are assumed to be faired. A complete moulded hullform is fair if all its hullform sections are fair.

3.7.40

hydrodynamic

forces resulting from the flow of liquid around a ship

3.7.41

hydrostatic

forces acting on the ship as a result of the pressure of air and water onto the outside under-water part of the ship when in calm water and not moving with respect to the fluid

3.7.42

implicit shape definition

the specification of the shape of a plate or stiffener part that is independent of any specific geometric modelling representation

NOTE It is defined by a minimal set of geometry elements and attributes such that an explicit shape could be derived from it.

3.7.43

interference shape

a collection of geometry defining the shape of a structural part for use in performing an interference analysis

NOTE This shape may include allowances for access and removal, insulation and shock clearances.

3.7.44

knuckle

one type of hullform element representing a mathematical 1st order (i.e., tangency) discontinuity between adjoining portions of the hullform, such as a chine

3.7.45

length between perpendiculars

the horizontal, longitudinal distance measured between the forward and aft perpendiculars

3.7.46

main machinery

a type of general characteristic that categorizes the elements that are contained in the class main machinery

NOTE The main machinery comprises all equipment used in primary systems (e.g., propulsion, steering and diving).

3.7.47

maintenance phase

the lifecycle stage when the ship is in operation and maintenance is carried out

3.7.48

manufacturer

a company or organization that makes or delivers parts or material for the ship

3.7.49

midship

a vertical transverse plane through the ship hull dividing the ship's length between perpendiculars into two

NOTE In some cases, the frame the nearest to this plane is assigned the designation amidship frame.

3.7.50

moulded hullform

the idealized, zero-thickness representation of a ship's hull

3.7.51

offset

the coordinate value of a point on the intersection line of an orthogonal plane with a ship moulded form

NOTE The offset is defined as the distance to the centreplane, or the distance to amidships if the plane lies in the longitudinal direction.

3.7.52

outfit and furnishing

a supertype of parts that represent non-structural and non-distributed system parts

NOTE The outfit and furnishing include such things as manholes, hatches, joiner bulkheads, ladders, gratings and windows.

3.7.53

predesign phase

the lifecycle phase when a first estimation of main characteristics for the ship are established, possibly based on previous designs

3.7.54

production engineering

the stage within the ship life-cycle when the ship is being built based on the production design

NOTE A subdivision can be made into structural part manufacturing and assembly manufacturing.

3.7.55

propulsion system

the system used to move the ship through the water

NOTE This can either be in a longitudinal (like the main propeller) or in a transverse (like bow-thruster) direction or a combination of both.

3.7.56

ratholes

the results of removing a portions of a structural part

NOTE The ratholes are usually in the shape of a semi-circle along one of its edges, in order to provide clearance between the part and a seam weld on an abutting part.

3.7.57

raw material stock

the plate and structural shape material that is purchased from either a steel mill or distributor, from which structural parts are cut

NOTE Typically, catalogues identify the range of sizes and cross-sectional shapes that can be selected. This raw material stock may be associated with a specific certification, heat treatment and chemical composition by the mill at the time of its manufacture.

3.7.58

reference geometry

one type of hullform element representing information symbolically defined in the hullform model and not intended to be an exact or complete geometric definition

NOTE Reference geometry may be used when the relationship between product model components is being stressed, not the geometric description.

EXAMPLE 1 The outline of the propeller blade or the shaft centerline for a moulded hullform.

3.7.59

sagging

the situation of a vessel in still water as well as in a seaway, in which a vessel is bending: the bottom of the ship would be in tension, the deck will be in compression

NOTE This is a reverse form of hogging with excess weight amidships.

3.7.60

sheer

the change in height of a sloping deck as it progresses from the bow to the stern

NOTE It is most commonly used on ships to counter the pitching motion effects by providing more freeboard near the ends of the vessel. Sheer for a deck is the measure on the centerline and is

always constrained to be linear. Each deck is limited to one sheer definition at the bow and one at the stern.

3.7.61

ship class

the grouping of multiple ships built from a single design, identified as similar by the use of a class name

3.7.62

ship offset

a point used as a co-ordinate for a location on a ship's hull form, appendage or deck

3.7.63

ship range

a measure of a ship's ability to remain at sea in order to perform a mission

NOTE It typically refers to a combination of distance and speed.

3.7.64

sight edge

one type of hullform element representing aspects of the moulded hullform that are highly visible to distant observers of the ship

NOTE For this reason, their shape is controlled, to a great extent, by appearance and aesthetic appeal.

3.7.65

special ship parts

ship parts that are formed by forging or casting processes

3.7.66

station

the intersection of a transverse plane with the moulded hullform

NOTE Typically the shape of the ship is defined by 20 stations equally spaced along the length of the vessel. These stations define the cross-sectional shape and are used along with the waterlines and buttocks to represent the 3-dimensional wireframe shape of the ship.

3.7.67

structural assembly

a hierarchical, building block view of individual structural parts comprising a ship

NOTE Parts are joined together to form successively larger building blocks. These interim products are referred to as assemblies.

3.7.68

structural connections and joints

the connectivity between structural parts and other structural parts where the geometry and topology of the connection is static and does not change as a result of the connection

NOTE A connection represents a requirement that two parts be joined and may decompose into lower level connection requirements. Eventually, a connection requirement is fulfilled by a joint which represents a physical realization of the connection requirement.

3.7.69

structural openings

an opening in a structural part to allow penetration by another structural part, penetration of a distribution system part, passage of air or liquid, access through a structural part, lightening of a structural part to improve structural efficiency, or clearance between joined structural parts

3.7.70

structural plate

a type of structural shape part that forms a separation between compartments

NOTE It is defined to lie on a ship surface.

3.7.71

structural stiffener

a type of structural shape part that serves to locally stiffen a structural plate part

NOTE Hence, it must be defined to lie on a ship surface. As such, the surface normal and the trace tangent are used to establish the plane for the cross-section. The cross-section is always placed on this plane based on the occurrence's vertex/point.

3.7.72

superstructure

a decked-over structure above the upper deck

NOTE the outboard sides of a superstructure are formed by the shell plating, as distinguished from a deckhouse that does not extend outboard to the shell plating.

3.7.73

system penetrations

the intersection of a piping, HVAC or outfitting part with a structural part (i.e., non-structural system penetration) or the intersection of one structural part with another structural part (i.e., a structural system penetration)

NOTE Depending on the location, compensation may be required to be added to the structural part being penetrated to compensate for lost material.

3.7.74

tangent

a type of hullform element representing a path where portions of the moulded hullform meet such that their 1st derivative is constant

3.7.75

tanks

a type of cargo hold especially designed for carrying liquids

3.7.76

tumblehome

one type of moulded hullform feature that is a collection of measurements representing the amount the moulded hullform's beam narrows above the design waterline

NOTE Tumblehome is the opposite of flare.

3.7.77

waterline

the intersection line of the water's surface with the ship's hull when the ship is afloat

NOTE In general, the waterline may vary between the intersections of any horizontal plane with the moulded hullform.

3.7.78

weight of steel

the total weight of the structural steel parts that make up the ship

NOTE During the various lifecycle phases this may vary from an estimated value (design) to a measured value (construction).

3.8 Abbreviations

For the purposes of this part of ISO 10303, the following symbols and abbreviations apply:

AAM	application activity model
AIC	application interpreted construct
AIM	application interpreted model
ARM	application reference model
AP	application protocol
B-rep	boundary representation solid model
CAD	computer aided design
CAM	computer aided manufacture
CSG	constructive solid geometry

CPS	collective protective systems
HVAC	heating, ventilation and air conditioning
ICLL	IMO load line conversion
IMO	international maritime organization
m	metre
MARPOL	marine pollution
mm	millimetre
NC	Numerical Control
PICS	protocol implementation conformance statement
SI	système international
SOLAS	safety of life at sea
SWBM	still water bending moment
SWSF	still water shear force
UoF	units of functionality

4 Information requirements

This clause specifies the information required for ship structures.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTE 1 A graphical representation of the information requirements is given in annex G.

NOTE 2 The information requirements correspond to those of the activities identified as being within the scope of this application protocol in annex F.

NOTE 3 The mapping table specified in 5.1 shows how the integrated resources and application interpreted constructs are used to meet the information requirements of this application protocol.

4.1 Units of functionality

This subclause specifies the units of functionality for the Ship Structures application protocol. This part of ISO 10303 specifies the following units of functionality:

- class_approvals;
- configuration_management;
- definitions;
- design_loads;
- external_references;
- hull_cross_sections;
- integrated_resources;
- items;
- location_concept;
- product_structures;
- ship_general_characteristics;
- representations;
- shapes;

- ship_manufacturing_definitions;
- ship_materials;
- ship_measures;
- structural_features;
- structural_parts;
- structural_systems;
- welds.

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in 4.2.

4.1.1 class_approvals

The class_approvals UoF is used for the definition of classification approval of structural parts and systems in general and plates, profiles and hull cross sections in particular. Definitions may have a class approval status attached. Subtypes of this entity indicate the approval status.

The following application objects are used by the class_approvals UoF:

Major subdivisions of this International Standard are:

- Corrugated_part_approval;
- Corrugated_part_rejection;
- Hull_cross_section_approval;
- Hull_cross_section_rejection;
- Hull_cross_section_result;
- Plate_approval;
- Plate_rejection;
- Plate_renewal_definition;
- Profile_approval;
- Profile_rejection;

- Profile_renewal_definition;
- Reason_for_decision;
- Structural_class_approval;
- Structural_class_approval_result;
- Structural_class_conditional_approval;
- Structural_class_rejection.

4.1.2 configuration_management

The configuration_management UoF covers the characterization of a definition under these principal aspects: its approval, its controlled changes, and the identity and the relationship of different versions within a version history.

The approval information provide when, who and what has been approved and to what level of approval, as well as how approvals are related to each other. The controlled changes information describes when and who modified what definition. Also it describes the impact of the change in terms of whether or not other definitions are created, modified or deleted.

The version of definitions is subject to versioning and how different versions are related to each other to provide a version history.

The following application objects are used by the configuration_management UoF:

Major subdivisions of this International Standard are:

- Approval_event;
- Approval_history;
- Change;
- Change_definition;
- Change_impact;
- Change_plan;
- Change_realization;
- Change_request;
- Check;

- Envisaged_definition_creation;
- Event;
- Revision;
- Revision_with_context;
- Version_creation;
- Version_deletion;
- Version_history;
- Version_modification;
- Version_relationship;
- Versionable_object_change_event.

4.1.3 definitions

The definitions UoF describes the abstract concept for the definition of items, item structures and item relationships.

The following application objects are used by the definitions UoF:

Major subdivisions of this International Standard are:

- Definition;
- Definition_relationship;
- Design_definition;
- Design_requirement;
- Functional_definition;
- General_characteristics_definition;
- Library_definition;
- Library_design_definition;
- Library_manufacturing_definition;

- Manufacturing_definition;
- Parametric_definition.

4.1.4 design_loads

The design_loads UoF provides documentation on loads that are applied to ship structures to prove their strength. Both actual loads and loads required by classification rules may be specified.

EXAMPLE 1 Bending moments and shear forces in still water or wave conditions are typical design loads.

A corresponding approval is normally based on artificial load cases that are the basis for strength calculations.

The following application objects are used by the design_loads UoF:

Major subdivisions of this International Standard are:

- Bending_moment;
- Design_load;
- Design_still_water_bending_moment;
- Design_still_water_shear_force;
- Design_SWSF_values;;
- Design_vertical_wave_bending_moment;
- Design_vertical_wave_shear_force;
- Maximum_permissible_still_water_bending_moment;
- Maximum_permissible_SWBM_values;
- Maximum_permissible_still_water_shear_force;
- Maximum_permissible_SWSF_values;
- Shear_Force.

4.1.5 external_references

The external_references UoF provides the capability and mechanisms by which references can be made to information in other product model data set or refer to information outside a given data exchange or data sharing context, and defines constructs for the identification and reference of standards and documents defined in external libraries or outside of the scope of ISO 10303.

The following application objects are used by the external_references UoF:

Major subdivisions of this International Standard are:

- Boolean_property_value;
- BSU;
- Class_BSU;
- Document;
- Document_reference;
- Document_reference_with_address;
- Document_usage_constraint;
- External_instance_reference;
- External_reference;
- Integer_property_value;
- Library_element_reference;
- Logical_property_value;
- Number_property_value;
- Property_BSU;
- Property_value;
- Real_property_value;
- String_property_value;
- Supplier_BSU;

— Universal_resource_locator.

4.1.6 hull_cross_sections

The hull_cross_sections UoF specifies the two dimensional (transverse) view of the hull at any longitudinal position on the ships, which is relevant to longitudinal strength. For this, the ship is viewed as a beam with a length of about the length between the perpendiculars with cross sectional properties varying over this length.

The structural steel works, making up the transverse cross section, are defined in terms of hull structural elements. A hull structural element is a plate area, possibly stiffened and possibly with cutouts, constrained by an underlying geometry, to fulfill structural or other functional roles.

The following application objects are used by the hull_cross_sections UoF:

Major subdivisions of this International Standard are:

- Flare_area;
- Hull_cross_section;
- Hull_cross_section_design_definition.

4.1.7 integrated_resources

UoF integrated_resources includes the following resources used in Ship APs: Major subdivisions of this International Standard are:

- p41_resources;
- p42_resources;
- p43_resources.

These resources are a copy of the Part 41, 42 and 43 resources with a number of modifications as indicated in the schemas. The entities in these schemas are used or referenced in the schemas belonging to the UoFs of AP218, and not documented individually in this part of ISO 10303.

4.1.8 items

The items UoF deals with the generic product structures of a ship, reflected by the concept of so-called "items" as well as by the concept of item structures and item relationships in which instances of items are collected and related to each other in a well defined way.

The following application objects are used by the items UoF:

Major subdivisions of this International Standard are:

- Definable_object;
- Item;
- Item_relationship;
- Item_structure;
- Ship;
- Versionable_object.

4.1.9 location_concept

The location_concept UoF specifies the information to locate a ship, or any part of it, in a right handed 3D Cartesian axis system. Also, it specifies the information required to subdivide any axis into intervals so that they form the reference basis for points in the axis system.

A co-ordinate system is either the one and only global co-ordinate system of the product description and root in the hierarchy, or a local co-ordinate system. Any number of local co-ordinate systems may exist. Spacing positions may be defined for any of the three global co-ordinate system axes. In case, the underlying co-ordinate system is the global co-ordinate system, the local origin maybe defined with reference to spacing positions.

The following application objects are used by the location_concept UoF:

Major subdivisions of this International Standard are:

- Buttock_table;
- Frame_table;
- Global_axis_placement;
- Local_co_ordinate_system;
- Local_co_ordinate_system_with_position_reference;
- Longitudinal_position;
- Longitudinal_table;
- Spacing_position;
- Spacing_position_with_offset;

- Spacing_table;
- Station_table;
- Transversal_position;
- Transversal_table;
- Vertical_position;
- Vertical_table;
- Waterline_table.

4.1.10 product_structures

The product_structures UoF describes the product structures of a ship from different points of view. It includes the assembly view of the ship, necessary for both production purposes and block oriented design.

Also, this UoF represents the functional view on a ship in order to highlight the specific aspects on each system type, used in the early design and detail design stages, and also intensively used during classification of a ship. This UoF provides a hierarchical definition of the assemblies and systems representing any number of levels.

The following application objects are used by the product_structures UoF:

Major subdivisions of this International Standard are:

- Assembly;
- Assembly_relationship;
- Composite_feature;
- Explicit_feature_design_definition;
- Feature;
- Feature_design_definition;
- Feature_relationship;
- Part;
- Part_relationship;

- System;
- System_design_definition;
- System_relationship.

4.1.11 representations

UoF representations includes the following resources used in Ship APs: Major subdivisions of this International Standard are:

- representation_resources.

The representations schema provides the core model of the abstract representation concept. A representation encapsulates those data, which are dependent on an interpretable context. Geometry is, for example a context dependent representation of shape, and there may be multiple possible representations of a shape.

Representation is not limited to geometry though, but may be applied to any product aspect that might be described differently depending on context.

4.1.12 shapes

UoF shapes includes the following resources used in this AP: Major subdivisions of this International Standard are:

- geometry_model_resources;
- geometry_resources;
- p501_aic (edge_based_wireframe)
- p508_aic (non_manifold_surface)
- p510_aic (geometrically_bounded_wireframe)
- p511_aic (topological_bounded_surface)
- p514_aic (advanced_B-rep)
- topology_resources.

The geometry_model_resources schema provides entities from the 10303-42 geometry model schema. The geometry_resources schema provides entities from the 10303-42 geometry schema. The topology_resources schema provides entities from the 10303-42 topology schema. The purpose of these schemas is not to model types of solid, surface geometric models and topology

models available but to provide entities that complete reference chains in other ship structures schemas.

AICs part 501, 508, 510, 511 and 514 are used in this AP to define the shape of ship's structures. They are not documented individually in this part of ISO 10303.

4.1.13 ship_general_characteristics

The ship_general_characteristics UoF specifies the basic information that details the ship's dimension and identification. This information is independent of any geometric context. This information includes scalar values for principal dimensions of a ship, designation information for ship related companies, ship's class notation and all relevant rules and regulations. Also the information about the lightships weight distribution and free-board characteristics is included in this UoF.

The following application objects are used by the ship_general_characteristics UoF:

Major subdivisions of this International Standard are:

- Carrier;
- Class_and_statutory_designation;
- Class_notation;
- Class_parameters;
- Freeboard_characteristics;
- Lightship_definition;
- Lightship_weight_item;
- Loadline;
- Moment_3d;
- Navy_Ship;
- Owner_designation;
- Research_Ship;
- Principal_characteristics;
- Regulation;

- Ship_designation;
- Shipyard_designation;
- ShipType;
- Weight_and_centre_of_gravity,
- Working_Ship.

4.1.14 ship_manufacturing_definitions

The ship_manufacturing_definitions UoF specifies modifications to the detailed design geometry to account for part edge stock, part edge preparation, the addition of layout and marking information, and, in the case of curved or bent plate and structural shape parts, flat pattern development. The part geometries are associated with the raw stock or plate blankets from which they will be cut. Constructs are provided for the use of manual or Numeric Control (NC) methods for laying out and cutting the parts from their raw stock and NC attributes are associated with the layout and marking information. Any production templates necessary for fabricating the parts are also identified.

Also this UoF specifies the definitions needed by the assembly manufacturing process. The definitions are associated with the assembly orientation, assembly bounding box and footprint, and (or) the weight and the centre of gravity of an assembly. The mechanisms specified by the weld manufacturing definition will be executed during the manufacturing life cycle.

The following application objects are used by the ship_manufacturing_definitions UoF:

Major subdivisions of this International Standard are:

- Assembly_bounding_box;
- Assembly_manufacturing_position;
- Assembly_manufacturing_definition;
- Plate_manufacturing_definition;
- Profile_manufacturing_definition;
- Structural_added_material_boundary_relationship;
- Structural_added_material_feature;
- Structural_added_material_feature_design_definition;
- Structural_manufacturing_feature;

- Structural_part_manufacturing_definition;
- Structural_weld_shrinkage_allowance_feature;
- Structural_weld_shrinkage_allowance_feature_design_definition;
- Weld_manufacturing_definition;
- Weld_testing;
- Welding_sequence.

4.1.15 ship_materials

The ship_materials UoF describes the materials used for building a ship structure or structural parts. It specifies raw materials by physical properties, not a material that is a product which may contain many raw materials. The ship materials include composite materials and homogeneous materials.

The following application objects are used by the ship_materials UoF:

Major subdivisions of this International Standard are:

- Composite_ship_material_property;
- Homogeneous_ship_material_property;
- Ship_material_property.

4.1.16 ship_measures

The ship_measures UoF includes all resources that are required for representing measures for physical quantities.

The following concepts are in scope of the ship_measures UoF: Major subdivisions of this International Standard are:

- global representation of units;
- SI units;
- derived units (such as for speed);
- conversion based units (such as inch and foot);
- all measures and units needed for all the ship APs.

It is explicitly out of scope to assign couples of measures and units to individual items instead of only values. The fact that `measures_with_unit` are nevertheless included is only due to the need for `conversion_based_units`. The distinction between `si_unit` and `conversion_based_unit` is made by using the "and/or" relationship of the `named_unit` subtypes.

EXAMPLE 1 A `length_measure` is ISO conformant if its unit is both a `length_unit` and a `si_unit`; it is a non-standard unit if it is both a `length_unit` and a `conversion_based_unit`.

The following application objects are used by the `ship_measures` UoF:

Major subdivisions of this International Standard are:

- `Angular_stiffness_measure_with_unit`;
- `Angular_stiffness_unit`;
- `Area_measure_with_unit`;
- `Area_unit`;
- `Density_measure_with_unit`;
- `Density_unit`;
- `Derived_unit`;
- `Derived_Unit_Element`;
- `Dilatation_measure_with_unit`;
- `Dilatation_unit`;
- `Extended_conversion_based_unit`;
- `Extended_global_uncertainty_assigned_context`;
- `Extended_measure_with_unit`;
- `Extended_uncertainty_measure_with_unit`;
- `Flow_rate_measure_with_unit`;
- `Flow_rate_unit`;
- `Force_measure_with_unit`;
- `Force_unit`;

- Frequency_measure_with_unit;
- Frequency_unit;
- Heat_rate_measure_with_unit;
- Heat_rate_unit;
- Inertia_moment_measure_with_unit;
- Inertia_moment_unit;
- Length_measure_with_unit;
- Length_unit;
- Linear_stiffness_measure_with_unit;
- Linear_stiffness_unit;
- Luminous_intensity_measure_with_unit;
- Luminous_intensity_unit;
- Mass_measure_with_unit;
- Mass_unit;
- Moment_measure_with_unit;
- Moment_unit;
- Named_unit;
- Plane_angle_measure_with_unit;
- Plane_angle_unit;
- Pressure_measure_with_unit;
- Pressure_unit;
- Product_of_inertia_measure_with_unit;
- Product_of_inertia_unit;
- Ratio_measure_with_unit;

- Ratio_unit;
- Section_modulus_measure_with_unit;
- Section_modulus_unit;
- Si_unit;
- Speed_measure_with_unit;
- Speed_unit;
- Stress_measure_with_unit;
- Stress_unit;
- Thermodynamic_temperature_measure_with_unit;
- Thermodynamic_temperature_unit;
- Time_measure_with_unit;
- Time_unit;
- Voltage_measure_with_unit;
- Voltage_unit;
- Volume_measure_with_unit;
- Volume_unit.

4.1.17 structural_features

The structural_features UoF specifies the abstract specializations of product items covering definitions of structural cutouts. The structural cutouts specify the nature of cutouts, that is the removal of material from a structural part or a composite structural object, such as panels and assemblies.

Cutouts are distinguished by the number of part boundaries that they effect into: edge cutouts, 1 boundary effected corner cutouts, 2 boundaries effected and interior cutouts (holes), no boundary effected.

The following application objects are used by the structural_features UoF:

Major subdivisions of this International Standard are:

- Bevel_design_definition;
- Circular_cutout_design_definition;
- Corner_cutout;
- Corner_cutout_boundary_relationship;
- Corner_cutout_design_definition;
- Drain_hole_cutout_design_definition;
- Edge_cutout;
- Edge_cutout_functional_definition;
- Edge_feature;
- Edge_feature_functional_definition;
- Elliptical_cutout_design_definition;
- Elongated_oval_cutout_design_definition;
- Free_form_interior_cutout_design_definition;
- Interior_cutout;
- Interior_cutout_design_definition;
- Interior_cutout_functional_definition;
- Inward_round_corner_design_definition;
- Outward_round_corner_design_definition;
- Part_edge_cutout_design_definition;
- Position_feature;
- Position_feature_design_definition;
- Position_feature_relationship;
- Rectangular_cutback_corner_design_definition;
- Rectangular_cutout_design_definition;

- Round_corner_design_definition;
- Round_corner_rectangular_cutout_design_definition;
- Round_edge_rectangular_cutout_design_definition;
- Seam;
- Seam_curve_relationship;
- Seam_design_definition;
- Shear_bevel_design_definition;
- Structural_cutout;
- Structural_cutout_boundary_relationship;
- Structural_feature;
- Structural_feature_relationship;
- Structural_part_penetration_relationship;
- Structural_system_penetration_relationship;
- Triangular_cutout_design_definition.

4.1.18 structural_parts

The structural_parts UoF specifies abstract specializations of ship parts. A structural part is the generalization of piece parts in a steel structure such as plates and profiles (e.g., stiffeners and beams). A structural part may be a part of many different product structures such as assemblies, boundary topology relations, connections and functional decomposition. In addition to the inherited properties a structural part has mass and material properties defined. Plates are specializations of structural parts and a plate is a part of the realization of the hull. Profiles are specializations of structural parts with a bounding surface defined along a path perpendicular to the surface of the cross section.

The following application objects are used by the structural_parts UoF:

Major subdivisions of this International Standard are:

- Angle_bar_cross_section;
- Angle_profile_cross_section;

- Bar_profile_cross_section;
- Bulbflat_cross_section;
- Channel_cross_section;
- Channel_profile_cross_section;
- Circular_hollow_profile_cross_section;
- Corrugated_part;
- Corrugated_part_design_definition;
- Corrugation;
- Explicit_profile_cross_section;
- Flanged_Plate_cross_section;
- Flanged_profile_cross_section;
- Flat_bar_cross_section;
- Library_profile_cross_section;
- Non_circular_hollow_profile_cross_section;
- Parametric_profile_cross_section;
- Plate;
- Plate_boundary_relationship;
- Plate_design_definition;
- Plate_functional_definition;
- Plate_relationship;
- Profile;
- Profile_boundary_relationship;
- Profile_cross_section;
- Profile_curve_trace_line;

- Profile_design_definition;
- Profile_functional_definition;
- Profile_relationship;
- Profile_trace_line_relationship;
- Proprietary_profile_cross_section;
- Round_bar_cross_section;
- Section_properties;
- Square_tube_cross_section;
- Structural_part;
- Structural_part_design_definition;
- Structural_part_functional_definition;
- Structural_part_relationship;
- Structural_part_symmetry_relationship;
- T_bar_cross_section;
- Twist_location;
- W_shape_cross_section.

4.1.19 structural_systems

The structural_systems UoF concentrates on structural systems, with the system specialization panel. A panel is the realization of a hull element, such as different types of bulkheads, tanks, decks, bottoms, etc. Possibly it is defined as a stiffened plate area defined on a surface, and constrained by a closed bounding curve. The curve may be modified by features such as cutouts, and the inner structure can be affected by holes.

This UoF contains definitions not only for a predefined breakdown into commonly used abstractions, but also offers the possibility for more shipyard specific abstractions. It handles properties such as tightness. Also, it is possible to characterize systems as belonging to the superstructure, ship subdivision or deckhouse.

The following application objects are used by the structural_systems UoF:

Major subdivisions of this International Standard are:

- Built_profile;
- Corrugated_structure;
- Corrugated_structure_design_definition;
- Corrugated_structure_boundary_relationship;
- Corrugated_structure_functional_definition;
- Corrugated_structure_relationship;
- Panel_system;
- Panel_system_boundary_relationship;
- Panel_system_curve_boundary;
- Panel_system_design_definition;
- Panel_system_functional_definition;
- Panel_system_plane_boundary;
- Panel_system_relationship;
- Plate_strake;
- Plate_strake_functional_definition;
- Structural_system;
- Structural_system_adjacency_relationship;
- Structural_system_design_definition;
- Structural_system_functional_definition;
- Structural_system_relationship;
- Structural_system_symmetry_relationship.

4.1.20 support_resources

UoF support_resources includes the following resources used in Ship APs: Major subdivisions of this International Standard are:

- `date_time_resources`;
- `organization_resources`;
- `support_resources`.

The `date_time_resources` schema provides concepts to specify a date and time. The `organization_resources` schema defines requirements for the definition of data about persons, organisations and their address. The `support_resources` schema declares basic primitive types.

4.1.21 welds

The welds UoF contains the information requirements necessary to present the welded connectivity between structural parts. Two or more parts are joined together, these requirements are fulfilled by structural parts connection implementation which represents a physical realization of the connection.

The definitions of this UoF consist of weld design definition, welded joint design definition. The design definitions indicate the design requirements of the welds and welded joint to ensure that they satisfy the requirements of the manufacturing process.

The following application objects are used by the welds UoF:

Major subdivisions of this International Standard are:

- `Beveled_groove_weld`;
- `Butt_groove_weld`;
- `Continuous_fillet_weld`;
- `Electrode_chemical_composition`;
- `Fillet_weld`;
- `Groove_weld`;
- `Intermittent_fillet_weld`;
- `Spot_seam_weld`;
- `Structural_part_connection_implementation`;
- `Structural_part_joint`;
- `Structural_part_joint_design_definition`;

- Weld;
- Weld_design_definition;
- Weld_filler_material;
- Welded_joint;
- Welded_joint_design_definition.

4.2 Application objects

This subclause specifies the application objects for the Ship Structures application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

4.2.1 Angle_bar_cross_section

An Angle_bar_cross_section is a type of Angle_profile_cross_section (see 4.2.2) identified by an angular shape consisting of a web and a flange.

NOTE ISO 657-1 and 657-2 specify dimensions for some standard equal-leg and unequal-leg angles.

The data associated with a Angle_bar_cross_section are the following:

- depth;
- radius;
- thk;
- width.

4.2.1.1 depth

The width specifies the depth of the web including the thickness of the flange.

4.2.1.2 radius

The width specifies the radius at the connection between the web and the flange.

4.2.1.3 thk

The width specifies the thickness of the web and the flange.

4.2.1.4 width

The width specifies the width of the flange including the thickness of the web.

Figure 3 shows the angular shape and definition of geometric parameters.

4.2.2 Angle_profile_cross_section

An Angle_profile_cross_section is a type of Parametric_profil_cross_section (see 4.2.178) identified by an angular shape consisting of a web and a flange. An Angle_profile_cross_section may

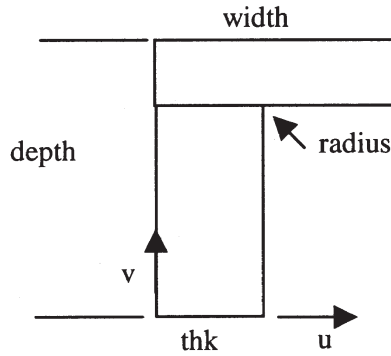


Figure 3 – An Angle_bar_cross_section

be one of the following: an Angle_bar_cross_section (see 4.2.1) or a T_bar_cross_section (see 4.2.292).

NOTE ISO 657-1 and 657-2 specify dimensions for some standard equal-leg and unequal-leg angles.

4.2.3 Angular_stiffness_measure_with_unit

An Angular_stiffness_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is the ability of resistance against angular movement.

4.2.4 Angular_stiffness_unit

An Angular_stiffness_unit is a type of Derived_unit (see 4.2.61) that is the unit of the force by which an angular movement is resisted.

4.2.5 Approval_event

An Approval_event is a type of Event (see 4.2.86) that triggers the change of a Approval_status.

The data associated with an Approval_event are the following:

- result;
- user_defined_result.

4.2.5.1 result

The result specifies the Approval_status the event leads to.

NOTE The Approval_status is the discrete qualification of an Approval in terms of its concretization.

The value of the result shall be one of the following:

- approved;
- rejected;
- unapproved;
- user defined.

NOTE See 4.2.5.1.1 - 4.2.5.1.4 for the definition of each allowable value for result.

4.2.5.1.1

approved

the Definition has been reviewed by the appropriate organization and is approved for use in the ship

4.2.5.1.2

rejected

the Definition has been reviewed by the appropriate organization and is not approved for use in the ship. One or more other Definitions would normally be created to replace the rejected Definition for the Item

4.2.5.1.3

unapproved

the Definition has not yet been reviewed or is in the process of being reviewed for approval by the organization

4.2.5.1.4

user defined

project-specific approval_status code to be determined by two or more exchanging organizations. A value of user defined for this type requires the population of the Optional User_defined_result attribute of Approval_event

4.2.5.2 user_defined_result

The user_defined_result specifies an user-defined approval status. This shall only be given when Approval_status.result is USER_DEFINED. The user_defined_result need not to be specified for a particular Approval_event.

4.2.6 Approval_history

An Approval_history collects all Approval_events (see 4.2.5) of a specific type defined for a Definition.

The data associated with an Approval_history are the following:

- approvals;
- status;
- subject.

4.2.6.1 approvals

The approvals specifies the sequence of Approval_events having occurred up to this point in time. Every Approval must have an associated Approval_status, thus the history must consist of at least one Approval_event. The list is assumed to be build up in chronological order. See 4.3.1 for the application assertion.

4.2.6.2 status

The status specifies the current approval status.

4.2.6.3 subject

The subject specifies the Definition this approval is related to.

NOTE A Definition may have zero, one or many associated Approvals. In case it has more then one associated Approval, all of them shall be different. See 4.3.2 for the application assertion.

4.2.7 Area_measure_with_unit

An Area_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a area as defined in ISO 31 (clause 2).

4.2.8 Area_unit

An Area_unit is a type of Named_unit (see 4.2.164) that is the unit in which the extent of a surface is measured.

4.2.9 Assembly

An Assembly is a type of both Item (see 4.2.132) and Item_structure (see 4.2.134). The assembly meta model is based on Item and on Item_structure (see 4.2.134) imposing an IMPLICIT hierarchical (tree) structure of assemblies from the view of manufacturing (e.g. unit assembly, preconstructed section). Assemblies can consist of (sub)Assembles or Parts. A WHERE rule assures that an Assembly is not (neither direct nor indirect) subassembly of itself. An other WHERE rule assures that no relationship in SELF.relationships is (neither direct nor indirect) between an Item of SELF.items and SELF. This means it is not allowed to have a relationship in a graph (SELF) between this graph and one of it's nodes (SELF.items). Nevertheless it is possible and allowed for SELF and one of it's nodes (SELF.items) to be related within an other context (graph). Assembly is NOT intended to be an ABSTRACT supertype to allow inter AP assemblies containing e.g. structural parts (AP218) and pipes (AP 217).

Figure 4 is an example of an Assembly.

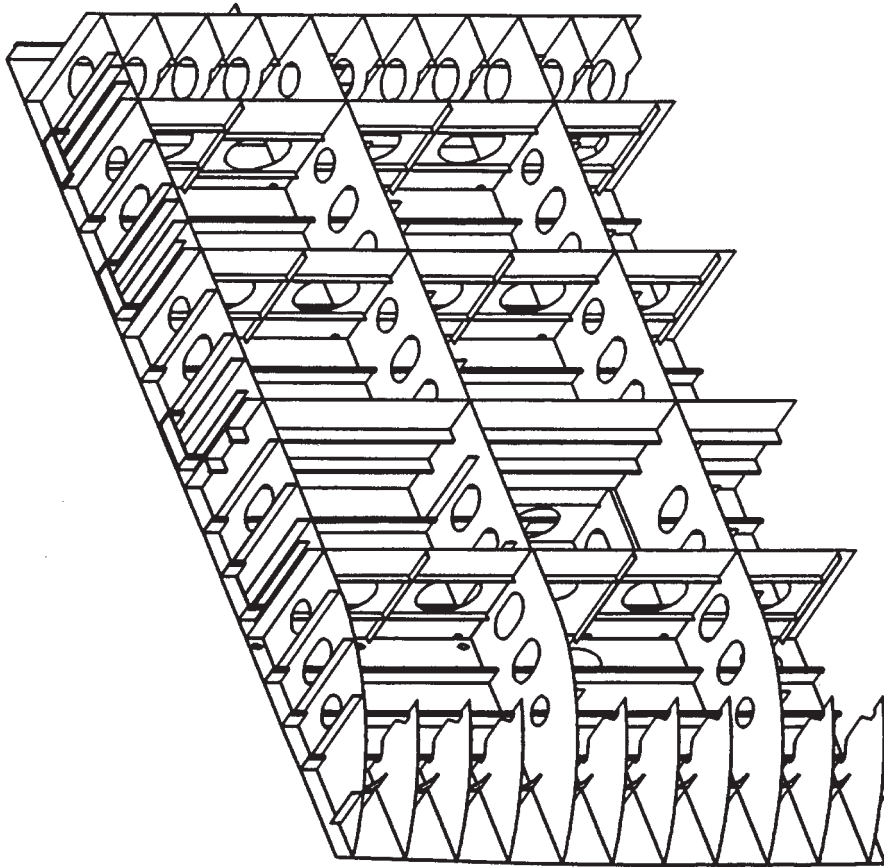


Figure 4 – An Assembly

The data associated with an Assembly are the following:

- class.

The class specifies the class of an Assembly in a verbal manner the range of the assembly tree where the Assembly is in. Each class may be one of the following: A `Predefined_assembly_class` type or a `Label` type. In the former case the value of the class shall be one of the following:

- preconstructed group;
- preconstructed section;
- production panel;
- unit assembly.

NOTE See 4.2.9.1.1 - 4.2.9.1.4 for the definition of each allowable value for class.

4.2.9.1

preconstructed group

is called also erection unit that can be consist of assemblies and potentially unit assemblies and individual parts, which when completed is erected onto the ship

4.2.9.2

preconstructed section

an assembly it may be consisting of part, unit assemblies and/or production panels, which when completed is assembled into an erection unit

4.2.9.3

production panel

a panel consisting of parts and/or unit assemblies, is defined within a production stage

4.2.9.4

unit assembly

an assembly consisting only of parts and no other assemblies, e.g., a built profile

4.2.10 Assembly_bounding_box

An `Assembly_bounding_box` specifies the cube circumscribing an `Assembly`. It is defined by two cartesian points for opposite corners of the bounding box. The edges of the cube are parallel to the co-ordinate axes.

The data associated with an `Assembly_bounding_box` are the following:

— `point_max`;

— `point_min`.

4.2.10.1 point_max

The `point_max` specifies the corner of the bounding box that has maximum co-ordinate values. See 4.3.3 for the application assertion.

4.2.10.2 point_min

The `point_min` specifies the opposite corner from the `point_max`, which has minimum co-ordinate values. See 4.3.3 for the application assertion.

4.2.11 Assembly_manufacturing_definition

An `Assembly_manufacturing_definition` is a type of `Manufacturing_definition` (see 4.2.154) that describes the definition of an `Assembly` (see 4.2.9) from the view of manufacturing.

The data associated with an `Assembly_manufacturing_definition` are the following:

- `assembly_drawing`;
- `assembly_method`;
- `assembly_positions`;
- `assembly_stage`;
- `centre_of_gravity`;
- `defined_for`;
- `mass`;
- `production_date`.

4.2.11.1 `assembly_drawing`

The `assembly_drawing` specifies the drawing on which the assembly is detailed for its planned stage and method of production. See 4.3.8 for the application assertion.

4.2.11.2 `assembly_method`

The `assembly_method` specifies the description of the technique to be used to assemble the Assembly.

4.2.11.3 `assembly_positions`

The `assembly_positions` define the set of positions that the defined assembly is placed in during production operations. See 4.3.6 for the application assertion.

4.2.11.4 `assembly_stage`

The `assembly_stage` specifies the production stage at which the Assembly is planned for assembly.

4.2.11.5 `centre_of_gravity`

The `centre_of_gravity` specifies the center of gravity of an Assembly defined by the `Assembly_manufacturing_definition`. See 4.3.6 for the application assertion.

4.2.11.6 `defined_for`

The `defined_for` specifies that `Assembly_manufacturing_definition` may only define Assemblies.

4.2.11.7 mass

The mass specifies the mass of an Assembly defined by the Assembly_manufacturing_definition. See 4.3.4 for the application assertion.

4.2.11.8 production_date

The production_date specifies the planned start of the work defined by the stage and method of production represented for an Assembly by the Assembly_manufacturing_definition. See 4.3.7 for the application assertion.

4.2.12 Assembly_manufacturing_position

An Assembly_manufacturing_position specifies an orientation of the defined assembly during production operations for that assembly, and its resultant footprint and bounding box while in this orientation.

The data associated with an Assembly_manufacturing_position are the following:

- assembly_footprint;
- bounding_box;
- orientation.

4.2.12.1 assembly_footprint

The assembly_footprint specifies the shape of the downside of an Assembly as seen by the mechanism supporting the Assembly for the related Assembly_manufacturing_position. The bounded curve must be a closed curve: the start point and the end point of the Bounded_curve are identical. See 4.3.9 for the application assertion.

4.2.12.2 bounding_box

The bounding_box specifies the cube circumscribing the Assembly defined by the Assembly_manufacturing_definition whose z-axis points in the direction of orientations. See 4.3.10 for the application assertion.

4.2.12.3 orientation

The orientation specifies the vector pointing in the +z-axis direction relative to the global coordinate system. See 4.3.11 for the application assertion.

4.2.13 Assembly_relationship

An Assembly_relationship is a type of Item_relationship (see 4.2.133) that describes the association of an Assembly (see 4.2.9) and an item allowed to be member in an Assembly. Therefore the related items are restricted to be either of type Assembly or Part. A WHERE rule assures that at least one of the related items is of type Assembly. The relationship is realized by collecting all Part_relationships between item_1 and item_2 into a Derived attribute atomic_relationships.

NOTE a) if one item is an Assembly and the other item is a Part atomic_relationships contains the Part_relationships between this Part and all the Parts hold by the Assembly.

b) if both, item_1 and item_2 are Assemblies atomic_relationships contains all Part_relationships between Parts of the 1. Assembly and Parts of the 2. Assembly.

The data associated with an Assembly_relationship are the following:

— item_1.

4.2.13.1 item_1

The item_1 specifies the re-declaration of item_1 attribute inherited from Item_relationship in order to constrain them to be of type Assembly that is related to a Part or another Assembly. See 4.3.12 for the application assertion.

4.2.14 Bending_moment

A Bending_moment is a type of Design_load (see 4.2.64) that is one way to represent Design_loads, that is by the physical quantity moment.

Figure 5 shows the bending moment in wave.

4.2.15 Bar_profile_cross_section

A Bar_profile_cross_section is a type of Parametric_profile_cross_section (see 4.2.178). A Bar_profile_cross_section may be one of the followings: a Bulbflat_cross_section (see 4.2.21), a Flat_bar_cross_section (see 4.2.102), or a Round_bar_cross_section (see 4.2.228). A Bar_profile_cross_section is identified by a solid round, square, or rectangular shape.

NOTE ISO 1035 specifies dimensions for some standard Bars.

4.2.16 Bevel_design_definition

A Bevel_design_definition is a type of Corner_cutout_design_definition (see 4.2.45) that specifies the line (or surface) obtained when cutting (parts of) a corner with a line (or plane) not parallel with the edges (or two of three surfaces) of the corner. The local origin is placed at the corner that shall be removed and the two (three) axis points along the edges.

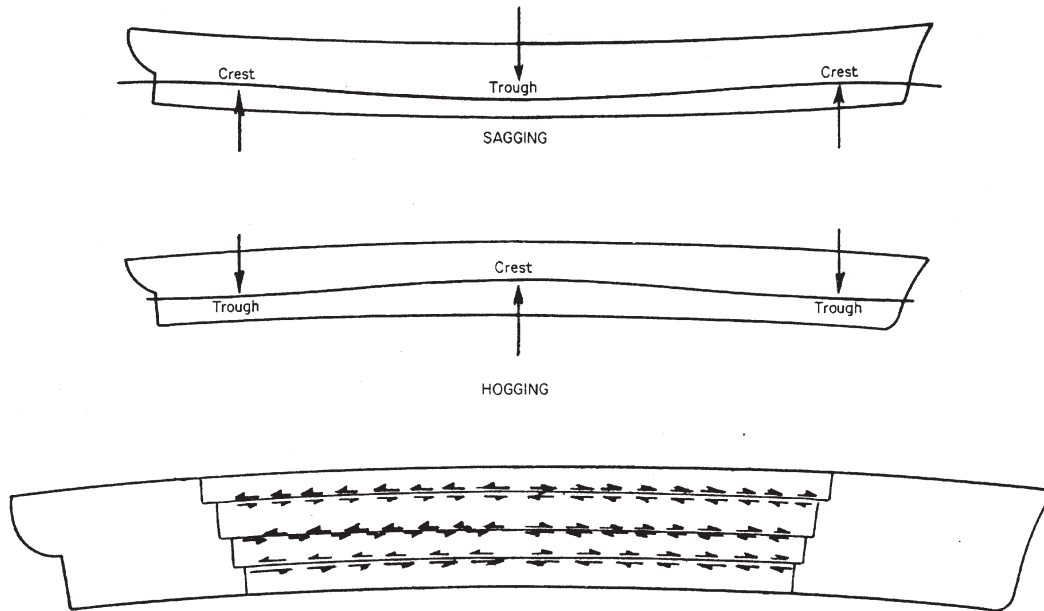


Figure 5 – Bedding moment

The data associated with a Bevel_design_definition are the following:

- offset;
- x_y_angle.

4.2.16.1 offset

The offset specifies the distance from local origin along x-axis.

4.2.16.2 x_y_angle

The x_y_angle specifies the angle between x-axis and bevelled side in the x-y-plane, see figure 6).

4.2.17 Beveled_groove_weld

A Beveled_groove_weld is a type of Groove_weld (see 4.2.115) where the parts being welded are of significantly different thickness and the edge or edges are trimmed so as to provide a smooth transition across the joint.

The data associated with a Beveled_groove_weld are the following:

- endcut_shape;

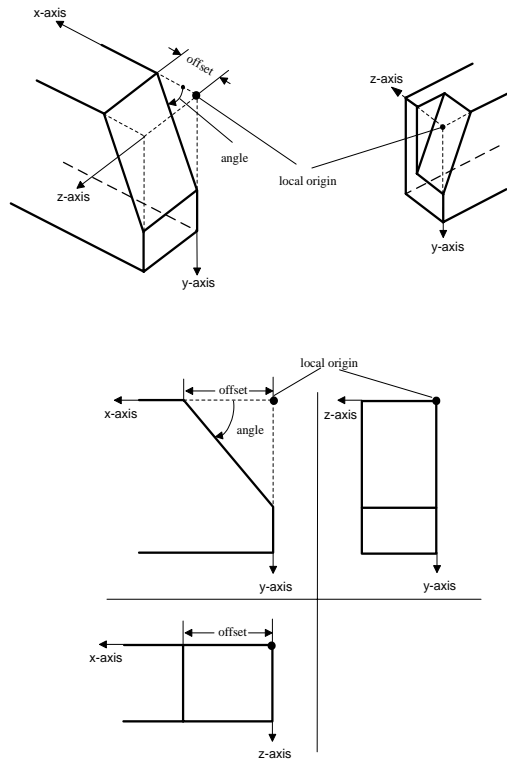


Figure 6 – Bevel_corner_cutout

- taper;
- taper_angle.

4.2.17.1 endcut_shape

The `endcut_shape` specifies an identifier used to denote the particular configuration of bevel being represented. The configuration includes the specification of whether the bevel groove weld has a flat, flare or curved faces.

The value of the `endcut_shape` shall be one of the following:

- double bevel;
- double J;
- double V;
- double U;

- flare double bevel;
- flare double V;
- flare single bevel;
- flare single V;
- single bevel;
- single J;
- single V;
- single U;
- user defined.

NOTE See 4.2.17.1.1 - 4.2.17.1.13 for the definition of each allowable value for `endcut_shape`.

4.2.17.1.1

double bevel

the bevel form which is non-symmetrically open and has a flat cut on one plate part, at both sides of the welded edge (called also K-form)

4.2.17.1.2

double J

the bevel form which is non-symmetrically open and has a curved cut on one plate part, at both sides of the welded edge

4.2.17.1.3

double U

the bevel form which is symmetrically open and has curved cut on both plate parts, at both sides of the welded edge

4.2.17.1.4

double V

the bevel form which is symmetrically open and has a flat cut on both plate parts, at both sides of the welded edge (called also X-form)

4.2.17.1.5

flare double bevel

the bevel form which is non-symmetrically open and spread outwards, has cut on one plate part, at both sides of the welded edge

4.2.17.1.6

flare double V

the bevel form which is symmetrically open and spread outwards, has cuts on both plate parts, at both sides of the welded edge

4.2.17.1.7**flare single bevel**

the bevel form which is non-symmetrically open and spread outwards, has cut on one plate part, at one side of the welded edge

4.2.17.1.8**flare single V**

the bevel form which is symmetrically open and spread outwards, has cuts on both plate parts, at one side of the welded edge

4.2.17.1.9**single bevel**

the bevel form which is non-symmetrically open and has a flat cut on one plate part, at one side of the welded edge

4.2.17.1.10**single J**

the bevel form which is non-symmetrically open and has a curved cut on one plate part, at one side of the welded edge

4.2.17.1.11**single U**

the bevel form which is symmetrically open and has curved cut on both plate parts, at one side of the welded edge

4.2.17.1.12**single V**

the bevel form which is symmetrically open and has a flat cut on both plate parts, at one side of the welded edge

4.2.17.1.13**user defined**

other possible bevel cut form of edge defined by the user

4.2.17.2 taper

The taper specifies an identifier used to denote the particular taper cut of bevel being represented. It includes the specification of whether the bevel groove weld has a taper cut, a one-side taper cut or a double-sides taper cut.

The value of the taper shall be one of the following:

- both sides taper;
- non taper;
- one side taper.

NOTE See 4.2.17.2.1 - 4.2.17.2.3 for the definition of each allowable value for taper.

4.2.17.2.1

both sides taper

both sides taper cut at the welded surface of a bevel weld

4.2.17.2.2

non taper

no taper cut at the welded surface of a bevel weld

4.2.17.2.3

one side taper

one side taper cut at the welded surface of a bevel weld

4.2.17.3 taper_angle

The taper_angle specifies the the measure of the taper angle. The taper_angle need not to be specified for a particular Beveled_groove_weld.

4.2.18 Boolean_property_value

A Boolean_property_value is a type of Property_value (see 4.2.216) that provides a value for a property of type BOOLEAN.

The data associated with a Boolean_property_value are the following:

- val.

The val specifies the value of type BOOLEAN.

4.2.19 BSU

A BSU (basic semantical unit) may be a Supplier_BSU (see 4.2.288), a Class_BSU (see 4.2.37), or a Property_BSU (see 4.2.215) that identifies a piece of information (which can be a supplier, a class or a property) by specifying a code and a version.

The data associated with a BSU are the following:

- code.

The code specifies the designation of the identification of the information piece.

4.2.20 Built_profile

A Built_profile is a type of both Structural_system (see 4.2.279) and Profile (see 4.2.203) that is made of at least two structural parts (e.g. Plates (see 4.2.184) or Profiles (see 4.2.203)) by an aggregation process; it can nevertheless be used like a profile.

The data associated with a Built_profile are the following:

- class.

The class specifies the class of a Built_profile needs to be BUILT_PROFILE.

4.2.21 Bulbflat_cross_section

A Bulbflat_cross_section is a type of Bar_profile_cross_section (see 4.2.15) consisting of a standard manufactured shape.

NOTE ISO 657-19 specifies dimensions for some standard bulb flats.

The data associated with a Bulbflat_cross_section are the following:

- bulb_radius;
- bulb_width;
- depth;
- tw.

Figure 7 shows the shape and definition of geometric parameters of a Bulbflat_cross_section.

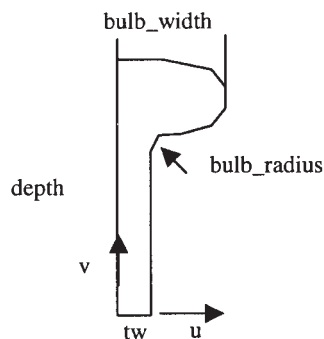


Figure 7 – A Bulbflat_cross_section

4.2.21.1 bulb_radius

The bulb_radius specifies the radius of bulb. See ISO 657-19. see xxx

4.2.21.2 bulb_width

The bulb_width specifies the bulb width. See ISO 657-19. see xxx

4.2.21.3 depth

The depth specifies the overall depth (height). See ISO 657-19. see xxx

4.2.21.4 tw

The tw specifies the thickness of the web. See ISO 657-19. see xxx

4.2.22 Butt_groove_weld

A Butt_groove_weld is a type of Groove_weld (see 4.2.115) where the parts being welded are of the same approximate thickness and no edge preparation (i.e. bevel or chamfer) is required.

The data associated with a Butt_groove_weld are the following:

— face_shape.

The face_shape specifies the face shape of parts being welded of a butt groove weld.

Figure 8 shows a one-side butt weld, figure 9 a both-sides butt weld.

4.2.23 Buttock_table

A Buttock_table is a type of Transversal_table (see 4.2.298) that is spacing table whose positions are a reference for the location of buttocks and are located on the global y axis.

4.2.24 Carrier

A Carrier is a type of a ShipType (see 4.2.243) that is a vessel which transports goods or passengers.

From the view of ship's structures Figure 10 shows a hull cross section of a bulk carrier, figure 11 shows its doublebottom.

The data associated with a Carrier are the following:

— has_type.

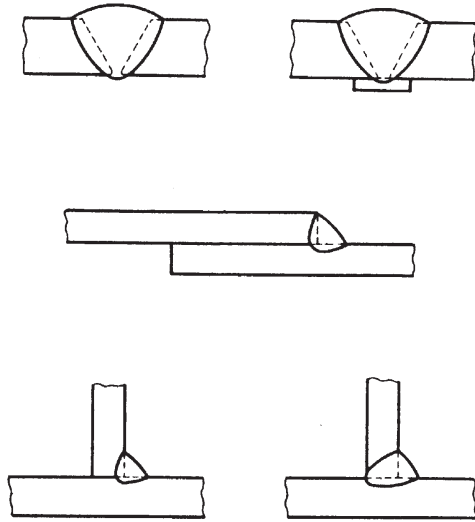


Figure 8 – One side butt weld

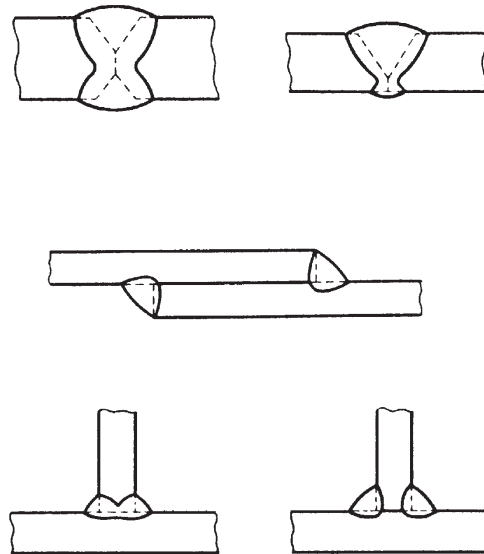


Figure 9 – Both sides butt weld

The `has_type` specifies the type of a Carrier.

The value of the `has_type` shall be one of the following:

— Barge;

- Barge Pontoon;
- Barge for Deck Loading;
- Barge for Liquefied Gas;
- Barge for Oil;
- Bulk Carrier;
- Car Carrier;
- Car Ferry;
- Cargo ship carrying passengers;
- Chemical Tanker;
- Chemical Tanker Type 1;
- Container Carrier;
- Cruise liner;
- Dry cargo Vessel;
- Ferry;
- Gas carrier;
- General Cargo Carrier;
- HighSpeedCraft Cargo;
- HighSpeedCraft Passenger;
- Liquefied Gas Tanker;
- Oil Tanker;
- Ore Carrier;
- Passenger Vessel;
- Product Tanker;
- Reefer;

- Refrigerated cargo carrying ship;
- Roro Vessel;
- Tanker for Refrigerated Fruit Juice;
- user defined.

NOTE See 4.2.24.1.1 - 4.2.24.1.29 for the definition of each allowable value for `has_type`.

4.2.24.1

Barge

a unmanned or manned vessel without self-propulsion, sailing in pushed or towed unit for the carriage of dry or liquid cargo

4.2.24.2

Barge Pontoon

a unmanned or manned floating unit with or without self-propulsion for the carriage of cargo

4.2.24.3

Barge for Deck Loading

a barge constructed for deck loading

4.2.24.4

Barge for Liquefied Gas

a barge for the carriage of liquid gas

4.2.24.5

Barge for Oil

a barge for the carriage of oil

4.2.24.6

Bulk Carrier

a cargo ship constructed for transportation of bulk

4.2.24.7

Car Carrier

a ship used for transportation of cars from manufacturer to end-user

4.2.24.8

Car Ferry

a cargo ship constructed for the transport of passengers, cars, coaches and lorries

4.2.24.9

Cargo ship carrying passengers

a cargo ship specially designed for carrying passengers

4.2.24.10

Chemical Tanker

a cargo ship specially designed for transportation of chemical

4.2.24.11

Container Carrier

a cargo ship constructed for the transport of containers

4.2.24.12

Cruise liner

a ship constructed for the transport of passengers for pleasure

4.2.24.13

Dry cargo Vessel

a cargo ship specially designed for dry cargo

4.2.24.14

Ferry

a cargo ship constructed for the transport of passengers, cars, coaches, lorries and trains. The ferry could be designed for only one type of cargo e.g., passengers

4.2.24.15

Gas carrier

a ship used for transportation of gas from manufacturer to end-user

4.2.24.16

General Cargo Carrier

a cargo ship constructed for the transport of any dry cargo which is not packed but carried loose such as grain or coal

4.2.24.17

HighSpeedCraft Cargo

a cargo ship constructed for the high-speed transport of cargo

4.2.24.18

HighSpeedCraft Passenger

a cargo ship constructed for the high-speed transport of passenger

4.2.24.19

Liquefied Gas Tanker

a cargo ship specially designed for transportation of chemicals

4.2.24.20

Oil Tanker

a cargo ship specially designed for transportation of oi

4.2.24.21**Ore Carrier**

a cargo ship constructed for transportation of Ore

4.2.24.22**Passenger Vessel**

a ship constructed for the transport of passengers

4.2.24.23**Product Tanker**

a cargo ship specially designed for transportation of oil products

4.2.24.24**Reefer**

a cargo ship constructed for the transport of any refrigerated cargo

4.2.24.25**Refrigerated cargo carrying ship**

a cargo ship specially designed for transportation of refrigerated cargo

4.2.24.26**Roro Vessel**

a cargo ship constructed for transportation of cargo, which can roll on board and roll off board

4.2.24.27**Tanker for Refrigerated Fruit Juice**

a cargo ship specially designed for transportation of refrigerated fruit juice

4.2.24.28**user defined**

a user defined carrier type

4.2.25 Change

A Change is a type of Item (see 4.2.132) that represents the focus of all stages associated with a potential or actual change to the product model resulting from a customer or design organization change order. The change may or may not result in modifications to the product model data. Any planned or actual changes to the product model are documented in the associated Change-definitions.

The data associated with a Change are the following:

— class.

The class specifies the qualification of the organizational role of the change.

EXAMPLE 1 Headquarter Modification Request or Engineering Change Proposal or such possible qualifications.

4.2.26 Change_definition

A `Change_definition` is a type of `Definition` (see 4.2.57) that is the generalization of the major discrete stages of a `Change`. A `Change_definition` is either a `Change_request` (see 4.2.30), a `Change_plan` (see 4.2.28), or a `Change_realization` (see 4.2.29).

The data associated with a `Change_definition` are the following:

- `author`;
- `date_time`;
- `defined_for`;
- `local_units`.

4.2.26.1 author

The `author` specifies the person and/or organization responsible for the change activities during the period lasting from the end of the previous (if it exists) up to the end of this `Change_definition`. See 4.3.15 for the application assertion.

4.2.26.2 date_time

The `date_time` specifies the date and time when this state was (approximately) reached. See 4.3.14 for the application assertion.

4.2.26.3 defined_for

The redefinition of the `defined_for` attribute, a `Change_definition` is only valid for one or multiple `Changes`. See 4.3.13 for the application assertion.

4.2.26.4 local_units

A `Change_definition` shall not define local units.

4.2.27 Change_impact

A `Change_impact` defines the effect a `Change` (see 4.2.25) will cause or has caused.

The data associated with a `Change_impact` are the following:

- `impact`.

The impact specifies the effect of a Change in terms of the creation, modification or deletion of some Definitions, Item_structures, or Item_relationships. See 4.3.16 for the application assertion.

4.2.28 Change_plan

A Change_plan is a type of Change_definition (see 4.2.26) that defines the proposed solution for a Change (see 4.2.25) . It is the basis for the activities, i.e., the Versionable_object_change_events (see 4.2.308) necessary to implement the Change in the product model.

The data associated with a Change_plan are the following:

- checks;
- chosen_solution_for;
- planned_impact.

4.2.28.1 checks

The checks specifies the checks planned for the Change. See 4.3.19 for the application assertion.

4.2.28.2 chosen_solution_for

The chosen_solution_for is the identification of the Change_request for which a Change_plan is applicable. See 4.3.18 for the application assertion.

4.2.28.3 planned_impact

The planned_impact specifies the estimated or calculated effects of the Change. This impact is usually chosen from the set of change_request.solution_alternatives. See 4.3.17 for the application assertion.

4.2.29 Change_realization

A Change_realization is a type of Change_definition (see 4.2.26) that defines the actual, observed effects of a Change (see 4.2.25).

The data associated with a Change_realization are the following:

- checks;
- impact;
- realization_of.

4.2.29.1 checks

The checks specifies the organizational approval of the product model revisions made to implement the Change. See 4.3.22 for the application assertion.

4.2.29.2 impact

The impact specifies the identification of the revisions made to the product model. See 4.3.20 for the application assertion.

4.2.29.3 realization_of

The realization_of specifies the Change_plan for which a product model change is being implemented. See 4.3.21 for the application assertion.

4.2.30 Change_request

A Change_request is a type of Change_definition (see 4.2.26) that is the first phase of a Change (see 4.2.25), where the need for a Change and possible solution_alternatives are established.

The data associated with a Change_request are the following:

- addressee;
- initiator;
- problem;
- solution_alternatives;
- solution_description.

4.2.30.1 addressee

The addressee specifies the person and/or organization the request is addressed to. The addressee need not to be specified for a particular Change_request. See 4.3.24 for the application assertion.

4.2.30.2 initiator

The initiator specifies the person and/or organization the request is coming from.

4.2.30.3 problem

The problem specifies the textual description of the problem having induced the request.

4.2.30.4 solution_alternatives

The solution_alternatives specifies the alternative solutions envisaged to each solve the problem. A solution is described in terms of the effect on versionable objects. See 4.3.23 for the application assertion.

4.2.30.5 solution_description

The solution_description specifies the textual description of one or more possible solutions for the problem. This textual description should be present, if the solution_alternatives are not yet established, or may enhance the information provided by the solution_alternatives. The solution_description need not to be specified for a particular Change_request.

4.2.31 Channel_cross_section

A Channel_cross_section is a type of Channel_profile_cross_section (see 4.2.32) that provides the parameters for the Channel shape.

NOTE ISO 657-11 specifies dimensions for some standard Channels.

The data associated with a Channel_cross_section are the following:

- depth;
- flange_thk;
- k;
- radius;
- web_thk;
- width.

4.2.31.1 depth

The width specifies the depth of the Channel_cross_section.

4.2.31.2 flange_thk

The flange_thk specifies the flange thickness of the Channel_cross_section.

4.2.31.3 k

The k specifies the distance from outside flange to fillet on web.

4.2.31.4 radius

The width specifies the radius of the Channel_cross_section.

4.2.31.5 web_thk

The width specifies the web thickness of the Channel_cross_section.

4.2.31.6 width

The width specifies the width of the Channel_cross_section.

4.2.32 Channel_profile_cross_section

A Channel_profile_cross_section is a type of Parametric_profil_cross_section (see 4.2.178). A Channel_profile_cross_section may be an Channel_cross_section. A Channel_profile_cross_section is identified by a Channel shape, consisting of a web and two flanges.

NOTE ISO 657-11 specifies dimensions for some standard Channels.

4.2.33 Check

A Check is a type of Event (see 4.2.86) that defines the details of a planned or fulfilled approval within an organization for a Change_plan (see 4.2.28) or a Change_realization (see 4.2.29).

4.2.34 Circular_cutout_design_definition

A Circular_cutout is a type of Interior_cutout_design_definition (see 4.2.128) that is given by the formula $x^2 + y^2 = r^2$, with r =radius as the only parameter representing the radius of the circular opening.

The data associated with a Circular_cutout_design_definition are the following:

— radius.

The radius specifies The radius that defines the shape of this circular cutout, see figure 12.

4.2.35 Circular_hollow_profile_cross_section

A Circular_hollow_profile_cross_section is a type of Parametric_profil_cross_section (see 4.2.178) that is circular in shape and has a non-solid core.

NOTE ISO 657-14 specifies dimensions for some standard circular hollow sections.

EXAMPLE 1 Pipe used as a structural stanchion would be represented using a Circular_hollow_profile_cross_section.

The data associated with a `Circular_hollow_profile_cross_section` are the following:

- `inner_diameter`;
- `outer_diameter`.

4.2.35.1 inner_diameter

The `inner_diameter` specifies the maximum dimension across the inner void.

4.2.35.2 outer_diameter

The `outer_diameter` specifies the maximum dimension between the outer faces.

4.2.36 Class_and_statutory_designation

A `Class_and_statutory_designation` is a type of `General_characteristics_definition` (see 4.2.113) that specifies the identification given to the ship by the classification society for the purpose of design, manufacture and in service approval. The data associated with a `Class_and_statutory_designation` are the following:

- `class`;
- `class_number`;
- `local_unit`;
- `statutory`.

4.2.36.1 tclass

The `class` specifies the applicable `Class_notation` with information about the ship type and the cargo. See 4.3.25 for the application assertion.

4.2.36.2 class_number

The `class_number` specifies the classification society specific identifier to a ship.

4.2.36.3 local_unit

The `local_unit` specifies that the `Class_and_statutory_designation` shall not define local units.

4.2.36.4 statutory

The `statutory` specifies the set of national and international regulations and standards with which the ship is intended to comply. See 4.3.26 for the application assertion.

4.2.37 Class_BSU

A Class_BSU (class basic semantical unit) is a type of BSU (see 4.2.19) that identifies a class in a parts library.

The data associated with a Class_BSU are the following:

- code;
- delivered_by;
- version.

4.2.37.1 code

The code specifies the designation of the identification of the class

4.2.37.2 delivered_by

The delivered_by specifies the library supplier who delivers the library class, i.e. the library context. See 4.3.27 for the application assertion.

4.2.37.3 version

The version specifies the designation of the version of the information piece.

4.2.38 Class_notation

A Class_notation specifies the notations given to the ship's hull and machinery by the classification society as a result of its approval activities during the design, manufacture and in service maintenance of the ship.

The data associated with a Class_notation are the following:

- approval_required_for_heavy_cargo;
- approval_required_for_oil_cargo;
- approval_required_loading_unloading_around;
- approval_required_loading_unloading_grabs;
- class_notations_hull;
- class_notations_machinery;

- `class_society`;
- `ice_class_notation`;
- `service_area`;
- `service_factor`.

4.2.38.1 approval_required_for_heavy_cargo

The `approval_required_for_heavy_cargo` is optional and, if present, specifies the flag indicating whether or not approval for special strengthening for heavy cargoes is necessary. These notations are valid for bulk carriers to indicate the distribution of loads across the cargo holds.

The value of the `approval_required_for_heavy_cargo` shall be one of the following:

- HC;
- HC E;
- HC EA.

NOTE See 4.2.38.1.1 - 4.2.38.1.3 for the definition of each allowable value for `approval_required_for_heavy_cargo`.

4.2.38.1.1

HC

strengthened for heavy cargo. Heavy bulk cargo may be unevenly distributed among the cargo holds

4.2.38.1.2

HC E

strengthened for heavy cargo. In addition to HC a non-homogeneous loading condition with empty holds on full draught is approved. The approved combination of empty holds is added to the notation (Holds 2, 3, 5 empty)

4.2.38.1.3

HC EA

strengthened for heavy cargo. Any cargo hold may be empty at full draught. The approved combinations of empty holds are added to the notation ("Holds 2, 3, 5 empty" or "... empty")

4.2.38.2 approval_required_for_oil_cargo

The `approval_required_for_oil_cargo` specifies the flag indicating whether or not approval is required for the carriage of oil cargoes.

4.2.38.3 approval_required_loading_unloading_aground

The `approval_required_loading_unloading_aground` specifies the flag indicating whether or not approval for loading and unloading aground is necessary.

4.2.38.4 approval_required_loading_unloading_grabs

The `approval_required_loading_unloading_grabs` specifies the flag indicating whether or not approval for loading and unloading using grabs is necessary.

4.2.38.5 class_notations_hull

The `class_notations_hull` specifies the notation given to the ship hull by the classification society as a result of its approval activities on the hull.

4.2.38.6 class_notations_machinery

The `class_notations_machinery` specifies the notation given to the ship's machinery by the classification society as a result of its approval activities on that machinery.

4.2.38.7 class_society

The `class_society` specifies the name and organisational details of the classification society whose rules and regulations are being used to assess the ship. See 4.3.28 for the application assertion.

4.2.38.8 ice_class_notation

The `ice_class_notation` specifies the type of class notation given to the ship indicating the ice conditions in which the ship has been approved to. The `ice_class_notation` need not to be specified for a particular `Class_notation`.

4.2.38.9 service_area

The `service_area` describes the area or route in which the ship operates. This may include information about waterway, wave, whether and wind conditions.

4.2.38.10 service_factor

The `service_factor` is related to the area and the occurring waves in this area in which the ship operates. The `service_factor` should be in the range of 0.5 to 1.0. The `service_factor` need not to be specified for a particular `Class_notation`.

4.2.39 Class_parameters

The `Class_parameters` are a type of `General_characteristics_definition` (see 4.2.113) that specify the length and speed of the ship in accordance with Classification Society rules and statutory

regulations.

The data associated with a Class_parameters are the following:

- block_coefficient_class;
- design_speed_ahead;
- design_speed_astern;
- length_class;
- length_solas;
- scantlings draught.

4.2.39.1 block_coefficient_class

The block_coefficient_class specifies the ratio of the moulded displacement volume to the volume of a block that has its length equal to the length_class, its breadth equal to the moulded_breadth and its depth equal to the scantlings draught.

4.2.39.2 design_speed_ahead

The design_speed_ahead specifies the forward speed at which the ship is designed to operate.

4.2.39.3 design_speed_astern

The design_speed_astern specifies the reverse speed at which the ship is designed to operate.

4.2.39.4 length_class

The length_class specifies the length measurement for the ship that is defined in Classification Society Rules.

4.2.39.5 length_solas

The length_solas specifies the length measurement for the ship measured in accordance with the International Convention on the Safety of Life at Sea.

4.2.39.6 scantlings draught

The scantlings draught specifies the summer load draught used by the Classification Society in its calculations for structural integrity and strength.

4.2.40 Composite_feature

A Composite_feature is a type of Feature (see 4.2.95) that is built up of other Features.

NOTE Composite_feature is intended for two applications:

- 1) to model a complex Feature that is composed of a set of simple Features.
- 2) to model Features that are logically defined on an aggregated level like a panel, but physically implemented on the atomic parts constituting the composite object e.g. on the plates realizing the panel. A manhole logically defined on a panel can be decomposed into corner-cutouts on the plates realizing the panel. When such decomposition is done, knowledge about the affected connections (e.g. plate seems) is required. The decomposition needs not necessarily be defined before plating of the panel is frozen. If the plating is changed, the decomposition structure must be maintained!

A subtype of Feature is made a composite feature by instantiating it as a complex subtype of Feature, that is combining itself with this Composite_feature subtype.

The data associated with a Composite_feature are the following:

— composed_of.

The composed_of specifies the Features which this Feature is built up of.

4.2.41 Composite_ship_material_property

A Composite_ship_material_property is a type of Ship_material_property (see 4.2.242) that is used to specify the raw material of this class by its physical properties.

4.2.42 Continuous_fillet_weld

A Continuous_fillet_weld is a type of Fillet_weld (see 4.2.98) where the weld bead is uninterrupted over the entire length of the weld. Usually, this type of weld is associated with joints requiring a higher degree of tightness.

4.2.43 Corner_cutout

A Corner_cutout is a type of Structural_cutout (see 4.2.264) that affects at least two edges of a Structural_part (see 4.2.269). Many types of Corner_cutouts may be distinguished. These shall be specified using different corner cutout design definitions, such as:

- Round_corner_design_definition,
- Inward_round_corner_design_definition,
- Outward_round_corner_design_definition,
- Bevel_design_definition,

- Shear_bevel_design_definition,
- Rectangular_cutback_corner_design_definition.

4.2.44 Corner_cutout_boundary_relationship

A `Corner_cutout_boundary_relationship` is a type of `Structural_cutout_boundary_relationship` (see 4.2.265) that specifies the topological relationship between a `Corner_cutout` (see 4.2.43) and the boundary segment of a `Plate` (see 4.2.184) or `Panel_system` (see 4.2.170).

NOTE `Corner_cutout_boundary_relationship` is intended for `Corner_cutouts` that describe weld preparations along the major edges of `Plates` or `Panel_systems`.

The data associated with a `Corner_cutout_boundary_relationship` are the following:

- `boundary_index_2`;
- `item_1`.

4.2.44.1 boundary_index_2

The `boundary_index_2` specifies the the index of the other boundary in the boundary list of the `Panel_system_design_definition` or `Plate_design_definition`, depending on which one is referenced by `item_2`, that has the cutout. A corner involves two boundary segments.

4.2.44.2 item_1

The `item_1` specifies the relating item of the relationship shall be a `Corner_cutout`. See 4.3.29 for the application assertion.

4.2.45 Corner_cutout_design_definition

A `Corner_cutout_design_definition` is a type of `Feature_design_definition` (see 4.2.96) specifies the design definition for a cutout at the corner of a profile, plate, or structural system.

The data associated with a `Corner_cutout_design_definition` are the following:

- `defined_for`;
- `z_depth`.

4.2.45.1 defined_for

The `defined_for` specifies the `Corner_cutouts` that a `Corner_cutout_design_definition` shall be valid for.

4.2.45.2 z_depth

The `z_depth` specifies the extension of the `Corner_cutout` in z-direction. If no `z_depth` is specified, the cutout is along the entire z-direction. The `z_depth` need not to be specified for a particular `Corner_cutout_design_definition`.

NOTE The starting-point of a `Corner_cutout` is given by the origin of the `Local_co_ordinate_system` that always is assigned to a `Structural_cutout` (except if `Free_form_interior_cutout_design_definition` is used).

4.2.46 Corrugated_part

A `Corrugated_part` is a type of `Structural_part` (see 4.2.269) that is made from corrugated stock material.

4.2.47 Corrugated_part_approval

A `Corrugated_part_approval` is a type of `Structural_class_approval` (see 4.2.260) that confirms the approval of a `Corrugated_part_design_definition`.

The data associated with a `Corrugated_part_approval` are the following:

— `subject`.

The `subject` specifies the approved `Corrugated_part_design_definition`. See 4.3.30 for the application assertion.

4.2.48 Corrugated_part_design_definition

A `Corrugated_part_design_definition` is a type of `Structural_part_design_definition` (see 4.2.271) that describes the definition of a `Corrugated_part` (see 4.2.46) from the view of detailed design.

The data associated with a `Corrugated_part_design_definition` are the following:

- `border`;
- `defined_for`;
- `mirrored`;
- `offset`;
- `repetition`;
- `shape_aspect`;
- `thickness`.

4.2.48.1 border

The border specifies that the Corrugated_part is assumed to be the extrusion of the 'cross section' (made of a 'repetition' Corrugations) along the local x-axis limited by 2 boundaries; boundary can be one of

- a) a given surface, or
- b) a boundary.

of a Corrugated_structure. Each border may be one of the following: a Surface (see ISO 10303, Part 42) or a Corrugated_structure_boundary. See 4.3.32, 4.3.34 and 4.3.35 for the application assertion.

4.2.48.2 defined_for

The defined_for specifies that a Corrugated_part_design_definition shall only be valid for Corrugated_parts. See 4.3.31 for the application assertion.

4.2.48.3 mirrored

The mirrored describes if the Corrugation is mirrored with respect to it's definition.

4.2.48.4 offset

The offset specifies the offset information for the location of the Corrugation.

4.2.48.5 repetition

The repetition specifies the number of Corrugations that make this Corrugated_part.

4.2.48.6 shape_aspect

The shape_aspect specifies the shape defining aspect of the 'cross section' of a Corrugated_part. See 4.3.33 for the application assertion.

4.2.48.7 thickness

The thickness specifies the material thickness of the corrugated_part

NOTE If the Corrugated_part belongs to a Structural_system that got a Design_definition during the preliminary design stage (like Corrugated_structure) thickness should match with the one specified there. For some reason however it might be necessary that a Corrugated_part gets a thickness other than specified during preliminary design.

4.2.49 Corrugated_part_rejection

A `Corrugated_part_rejection` is a type of `Structural_class_rejection` (see 4.2.263) that confirms the approval of a `Corrugated_part_design_definition` is not approved. The reasons for rejection and existing requirements for the `Corrugated_part` are specified.

NOTE `Corrugated_part_rejection` may specify additional information clarifying the rule requirements for the `Corrugated_part`. A yard has several options for action; the thickness of the part may be increased, the material quality may be improved, another stiffening may be applied etc. Although it is not mandatory, a class society may suggest improvements in these areas. This may then be done using `Structural_class_rejection.proposed_alternative`.

The data associated with a `Corrugated_part_rejection` are the following:

— `subject`.

The `subject` specifies the rejected `Corrugated_part_design_definition`. See 4.3.36 for the application assertion.

4.2.50 Corrugated_structure

A `Corrugated_structure` is a type of `Structural_system` (see 4.2.279) that has the restriction to only consist of `Corrugated_parts` (see 4.2.46) and `Plates` (see 4.2.184).

The data associated with a `Corrugated_structure` are the following:

— `class`.

The `class` specifies the class of a `Corrugated_structure` needs to be a `CORRUGATED_STRUCTURE`.

Figure 13 shows the parameters for the corrugated structures.

4.2.51 Corrugated_structure_boundary_relationship

A `Corrugated_structure_boundary_relationship` is a type of `Corrugated_structure_relationship` (see 4.2.54) that defines a boundary of a `Corrugated_structure` (see 4.2.50) by referring to a "structural item". "Structural_item" can be one of the following:

`Corrugated_structure` the boundary is created by the intersection of the (implicitly or explicitly) given moulded surfaces of the two `Corrugated_structures`;

`Panel_system` the boundary is created by the intersection of the (implicitly or explicitly) given moulded surfaces of the `Corrugated_structure` and the `Panel_systems`;

`Seam` the `Seam` is the boundary.

The data associated with a `Corrugated_structure_boundary_relationship` are the following:

- displacement.

The displacement specifies the offset orthogonally off the referenced boundary.

4.2.52 Corrugated_structure_design_definition

The `Corrugated_structure_design_definition` is a type of `Structural_system_design_definition` (see 4.2.281) that describes the definition of a `Corrugated_structure` (see 4.2.50) for the life-cycle stage preliminary design.

The data associated with a `Corrugated_structure_design_definition` are the following:

- border;
- corrugation_width;
- defined_for;
- moulded_surface;
- thickness.

4.2.52.1 border

The border specifies the border of a `Corrugated_structure` is made of a number of (at least 3) boundaries. The order of the boundaries shall be significant but there is NO rule that requires end point of `boundary[n]` = start point of `boundary[n+1]` furthermore two boundaries may intersect or it might be necessary to (straight) elongate one or both boundaries in order to make them intersecting; boundary can be one of

- a) a line resulting from the intersection of the `Corrugated_structure`'s moulded surface with the moulded surface of a given `Structural_system`;
- b) a curve parallel to a boundary of a given `Structural_system` with a given displacement;
- c) a Seam.

4.2.52.2 corrugation_width

The `corrugation_width` specifies the proposed length of the corrugation of the corrugated parts belonging to the corrugated structure.

NOTE The swedge length specified here serves as a reference value for the length of the Corrugation of the `Corrugated_parts` defined for the `Corrugated_structure` referred to in a later stage of the design.

For some reason it may happen that the corrugation length of the `Corrugated_parts` differs from the one specified here.

The `corrugation_width` need not to be specified for a particular `Corrugated_structure_design_definition`.

4.2.52.3 defined_for

The `defined_for` specifies that the `Corrugated_structure_definitions` shall only be declared for `Corrugated_structures`. See 4.3.37 for the application assertion.

4.2.52.4 moulded_surface

The `moulded_surface` specifies the underlying moulded surface of the `Corrugated_structure` (if not specified the local x-y-plane is implicitly assumed as underlying moulded surface). The `moulded_surface` need not to be specified for a particular `Corrugated_structure_design_definition`.

4.2.52.5 thickness

The `thickness` specifies the proposed material thickness of the `Corrugated_structure`.

NOTE The thickness specified here serves as a reference value for the thickness of the `Corrugated_parts` and `Plates` defined for the `Corrugated_structure` referred to in a later stage of the design. For some reason it may happen that the thickness of the `Corrugated_parts` and `Plates` differs from the one specified here.

4.2.53 Corrugated_structure_functional_definition

A `Corrugated_structure_functional_definition` is a type of `Structural_system_functional_definition` (see 4.2.282) defines the function of a `Corrugated_structure` (see 4.2.50).

The data associated with a `Corrugated_structure_functional_definition` are the following:

— `defined_for`.

The `defined_for` specifies the a `Corrugated_structure_functional_definition` shall only be valid for `Corrugated_structures`. See 4.3.38 for the application assertion.

4.2.54 Corrugated_structure_relationship

A `Corrugated_structure_relationship` is a type of `Structural_system_relationship` (see 4.2.284) that describes the association of a `Corrugated_structure` (see 4.2.50) with another 'structural item'. 'Structural item' can be one of

a) `Structural_system`;

- b) Structural_part ;
- c) Structural_feature.

The data associated with a Corrugated_structure_relationship are the following:

- item_1.

The item_1 specifies the Corrugated_structure_relationship shall relate a Corrugated_structure with something else. See 4.3.39 for the application assertion.

Figure 14 shows a corrugated bulkhead.

4.2.55 Corrugation

A Corrugation is the shape defining aspect of the 'cross section' of a Corrugated_part (see 4.2.46).

The data associated with a Corrugation are the following:

- depth;
- flat_width_1;
- flat_width_2;
- radius_1;
- radius_2;
- slope_width.

4.2.55.1 depth

The depth specifies the distance between the lower and the upper flat part of the corrugation shape.

4.2.55.2 flat_width_1

The flat_width_1 specifies the width of the lower flat part of the corrugation shape.

4.2.55.3 flat_width_2

The flat_width_2 specifies the width of the upper flat part of the corrugation shape.

4.2.55.4 radius_1

The radius_1 specifies the radius of the bend between the lower flat part and the slope part of the corrugation. The radius_1 need not to be specified for a particular Corrugation.

4.2.55.5 radius_2

The radius_2 specifies the radius of the bend between the upper flat part and the slope part of the corrugation. The radius_2 need not to be specified for a particular Corrugation.

4.2.55.6 slope_width

The slope_width specifies the width of the slope part of the corrugation shape. Figure 15 shows the parameters for the corrugation.

4.2.56 Definable_object

A Definable_object is anything that can be defined, i.e. that can be pointed to by Definitions.

The data associated with a Definable_object are the following:

- id.

The id specifies the global unique identifier for itself. See 4.3.40 for the application assertion.

4.2.57 Definition

A Definition is a type of Versionable_object (see 4.2.307) that is the supertype for all kinds of definitions for a Definable_objects (see 4.2.56).

The data associated with a Definition are the following:

- defined_for;

- id;

- local_units.

4.2.57.1 defined_for

The defined_for specifies the Definable_objects which are defined by Definition. See 4.3.41 for the application assertion.

4.2.57.2 id

The id specifies the global unique identifier for SELF. See 4.3.42 for the application assertion.

4.2.57.3 local_units

The local_units specifies the If the units that definition makes use of differ from the ones globally defined for the ship, they should be specified here.

4.2.58 Definition_relationship

A Definition_relationship provides the possibility to relate two Definitions (see 4.2.57).

The data associated with a Definition_relationship are the following:

- definition_1;
- definition_2;
- description.

4.2.58.1 definition_1

The definition_1 specifies the relating entities are only allowed to be of type Definition. See 4.3.43 for the application assertion.

4.2.58.2 definition_2

The definition_2 specifies the related entities are only allowed to be of type Definition. See 4.3.43 for the application assertion.

4.2.58.3 description

The description specifies The description, if required, for this relationship. The description need not to be specified for a particular Definition_relationship.

4.2.59 Density_measure_with_unit

A Density_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is the density of something.

4.2.60 Density_unit

A Density_unit is a type of Derived_unit (see 4.2.61) that is the unit of density of something, that is mass per volume.

4.2.61 Derived_unit

A Derived_unit is an expression of units.

The data associated with a `Derived_unit` are the following:

- `elements`.

The `elements` specifies the group of units and their exponents that define the `Derived_unit`. See 4.3.44 for the application assertion.

4.2.62 `Derived_unit_element`

A `Derived_unit_element` is one of the unit quantities which makes up a `Derived_unit`.

The data associated with a `Derived_unit_element` are the following:

- `exponent`;
- `unit`.

4.2.62.1 `exponent`

The `exponent` specifies the power that is applied to the `unit` attribute.

4.2.62.2 `unit`

The `unit` specifies the fixed quantity which is used as the mathematical factor. See 4.3.45 for the application assertion.

4.2.63 `Design_definition`

A `Design_definition` is a type of `Definition` (see 4.2.57) that is the supertype for all kinds of design definitions.

The data associated with a `Design_definition` are the following:

- `representations`.

The `representations` specify the representations of the design definition it is possible for a `Design_definition` to have multiple `Representations`. See 4.3.46 for the application assertion.

4.2.64 `Design_load`

A `Design_load` is a type of `Design_definition` (see 4.2.63) that is a load applied to the structure of a ship to prove its strength. The `Design_load` may be induced by an artificial load, such as a load required by a classification society, or an actual load, such as cargo. `Design_loads` may be expressed as moments or forces.

The data associated with a `Design_load` are the following:

— `defined_for`

The `defined_for` specifies the Ships that a `Design_load` may be defined for. See 4.3.47 for the application assertion.

4.2.65 `Design_requirement`

A `Design_requirement` is a type of `Definition` (see 4.2.57) that is created for the related ship exists in the heads of the designers only. Therefore this information is not specific with respect to product item, because they are not yet defined, but it is very specific with respect to the expectations the future owner and the designers would like to express.

The data associated with a `Design_requirement` are the following:

— `specification`.

The `specification` specifies the `Specification` documenting the design requirement. See 4.3.48 for the application assertion.

4.2.66 `Design_still_water_bending_moment`

A `Design_still_water_bending_moment` is a type of `Bending_moment` (see 4.2.14) that is the bending moment amidship that the ship is designed for in still water. The values may be based on actual cargo, ballast conditions, and sea-going conditions. If these are not known, standard moments given by classification rules shall be applied.

The data associated with a `Design_still_water_bending_moment` are the following:

— `hogging_amidship`;

— `sagging_amidship`.

4.2.66.1 `hogging_amidship`

The `hogging_amidship` specifies the hogging moment amidship.

4.2.66.2 `sagging_amidship`

The `sagging_amidship` specifies the sagging moment amidship.

Figure 16 shows `Design_still_water_bending_moment`.

4.2.67 `Design_still_water_shear_force`

A `Design_still_water_shear_force` is a type of `Shear_force` (see 4.2.239) that is a collection of shear force values distributed along the ship. These represent the shear force load that the ship

is designed for when being in sea-going condition.

The data associated with a `Design_still_water_shear_force` are the following:

- `table_of_shear_force_values`.

The `table_of_shear_force_values` specifies the set of values that defines the shear force distribution along the ship for design purposes. See 4.3.49 for the application assertion.

4.2.68 `Design_SWSF_values`

The `Design_SWSF_values` are design values for shear forces at specified positions along the ship. The values shall represent classification society requirements.

The data associated with a `Design_SWSF_values` are the following:

- `negative_value`;
- `position`;
- `positive_value`.

4.2.68.1 `negative_value`

The `negative_value` specifies the negative shear force value that contributes to overall shear.

4.2.68.2 `position`

The `position` specifies the distance from the aft-perpendicular from where the shear forces are taken, given as a frame or station position. See 4.3.50 for the application assertion.

4.2.68.3 `positive_value`

The `positive_value` specifies the positive shear force value that contributes to overall shear.

4.2.69 `Design_vertical_wave_bending_moment`

A `Design_still_water_bending_moment` is a type of `Bending_moment` (see 4.2.14) that is the bending moment amidship that the ship is designed for being in sea-going conditions. Both hogging and sagging moments have distributions over the length of the ship. These can be derived from the values amidships and the length given by the classification society rules. The values will change for short voyages and sheltered water conditions. These can be derived from the sea-going conditions and so are not necessary for a model used for exchange purposes.

The data associated with a `Design_vertical_wave_bending_moment` are the following:

- hogging;
- sagging.

4.2.69.1 hogging

The hogging specifies the hogging moment amidships.

4.2.69.2 sagging

The sagging specifies the sagging moment amidships.

4.2.70 Design_vertical_wave_shear_force

A Design_vertical_wave_shear_force is a type of Shear_force (see 4.2.239) that is the shear force amidship that the ship is designed for being in sea-going conditions. Both positive and negative values have distributions over the length of the ship. These can be derived from the values amidships and the length given by the classification society rules. The values will change for short voyages and sheltered water conditions. These can be derived from the sea-going conditions and so are not necessary for a model used for exchange purposes.

The data associated with a Design_vertical_wave_shear_force are the following:

- negative_value;
- positive_value.

4.2.70.1 negative_value

The negative_value specifies the negative shear force value amidships.

4.2.70.2 positive_value

The positive_value specifies the positive shear force value amidships.

4.2.71 Dilatation_measure_with_unit

A Dilatation_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is the dilatation of something.

4.2.72 Dilatation_unit

A Dilatation_unit is a type of Derived_unit (see 4.2.61) that is the unit of the expansion of material due to change of temperature, that is length per temperature.

4.2.73 Document

A Document is a type of Versionable_object (see 4.2.307) that is an unambiguous identification of some human readable data item defined outside ISO 10303. A document has an author and may be versioned.

The data associated with a Document are the following:

- has_author;
- has_title;
- source_type;
- summary.

4.2.73.1 has_author

The has_author specifies the person and/or organization that authored the Document.

4.2.73.2 has_title

The has_title specifies that each Document has a descriptive title consisting of a word, or group of words.

4.2.73.3 source_type

The source_type specifies the type of the document (i.e. printed document, file) One should think about what other kinds of source_types are possible and if there should be better an enumeration type for this attribute.

4.2.73.4 summary

The summary specifies the summary or abstract, gives a short textual description of the content of a document. The summary need not to be specified for a particular Document.

4.2.74 Document_reference

A Document_reference defines the qualification of a Document in terms of its source or location. If source is i.e. a book, the pointer could be a section label or a page number.

The data associated with a Document_reference are the following:

- assigned_document.

The assigned_document specifies the An assigned_document specifies the document (or portion of a document) which is to be associated with the product data.

4.2.75 Document_reference_with_address

A Document_reference_with_address is a type of both Document_reference (see 4.2.74) and External_reference (see 4.2.94) with a pointer to a location inside the source.

4.2.76 Document_usage_constraint

A Document_usage_constraint defines the applicability of a Document (see 4.2.73). Applicability may be defined in terms of selecting a specific part of a document and/or interpreting the content of that specific part.

The data associated with a Document_usage_constraint are the following:

- element_name;
- line_number;
- page;
- paragraph;
- section;
- source.

4.2.76.1 element_name

The element_name specifies the name for this subset of the document.

4.2.76.2 line_number

The line_number specifies the line number. The line_number need not to be specified for a particular Document_usage_constraint.

4.2.76.3 page

The page specifies the page number. This may not always be a plain number, but also Roman number or a combination of chapter and page number. The page need not to be specified for a particular Document_usage_constraint.

4.2.76.4 paragraph

The paragraph specifies the paragraph identifier. The paragraph need not to be specified for a particular Document_usage_constraint.

4.2.76.5 section

The section specifies the section label. The section need not to be specified for a particular Document_usage_constraint.

4.2.76.6 source

The source specifies the source defines, to which document the following specification of sections, pages, paragraphs is related. See 4.3.51 for the application assertion.

4.2.77 Drain_hole_cutout_design_definition

A Drain_hole_cutout_design_definition is a type of Feature_design_definition (see 4.2.96). A drain hole is a symmetric oblong opening with circular ends on both sides and a break-through along the symmetry line to the exterior of the work item. The drain holes symmetry axis is perpendicular to the work item boundary. It points into the interior of the work item. The origin of the local co-ordinate system is placed at the intersection of the work item boundary line and the symmetry axis. All length measures (width, depth, drain_hole_radius, gap and gap_radius) are always positive. Depth is measured perpendicular to the item edge (y-axis = symmetry axis) whereas width and gap is parallel to the item edge (x-axis).

The data associated with a Drain_hole_cutout_design_definition are the following:

- defined_for;
- depth;
- drain_hole_radius;
- gap;
- gap_radius;
- width.

4.2.77.1 defined_for

The defined_for specifies the Edge_cutouts that a Drain_hole_cutout_design_definition shall be valid for.

4.2.77.2 depth

The depth specifies the distance measured from the exterior of the work item through the gap to the furthest side of the drain hole. See 4.3.52 for the application assertion.

4.2.77.3 drain_hole_radius

The `drain_hole_radius` specifies the radius of the circle piece that determines the round shape of the drain hole.

4.2.77.4 gap

The `gap` specifies the width of the gap.

4.2.77.5 gap_radius

The `gap_radius` specifies the radius of the round corner that represents the transition from the drain hole to the gap.

4.2.77.6 width

The `width` specifies the maximum diameter of the drain hole.

See figure 17 for a `Drain_hole_cutout`.

4.2.78 Edge_cutout

An `Edge_cutout` is a type of `Structural_cutout` (see 4.2.264) that is the feature along an edge of a `Structural_part` (see 4.2.269). An `Edge_cutout` specifies the removal of material for a certain purpose, such as for the penetration of another `Structural_part` or for water escapes. Many types of `Edge_cutouts` may be distinguished. These shall be specified using different edge cutout design definitions, such as:

- `Part_edge_cutout_design_definition`,
- `Drain_hole_cutout_design_definition`.

4.2.79 Edge_cutout_functional_definition

An `Edge_cutout_functional_definition` is a type of `Functional_definition` (see 4.2.112) that defines the function that an `Edge_cutout` (see 4.2.78) may be designed for.

The data associated with an `Edge_cutout_functional_definition` are the following:

- `defined_for`;
- `function`.

4.2.79.1 defined_for

The `defined_for` specifies the `Edge_cutout` feature that this function is assigned to. See 4.3.53 for the application assertion.

4.2.79.2 function

The function specifies the functionality of the `Edge_cutout` that this definition points to.

The value of the function shall be one of the following:

- access hole;
- air escape;
- air liquid escape;
- liquid escape;
- penetration;
- user defined.

NOTE See 4.2.79.2.1 - 4.2.79.2.6 for the definition of each allowable value for function.

4.2.79.2.1

access hole

the `Edge_cutout` is used as an access hole

4.2.79.2.2

air escape

the `Edge_cutout` is used as an air escape

4.2.79.2.3

air liquid escape

the `Edge_cutout` is used as an air and liquid escape

4.2.79.2.4

liquid escape

the `Edge_cutout` is used as a liquid escape

4.2.79.2.5

penetration

the `Edge_cutout` is used as a penetration

4.2.79.2.6

user defined

the function code for the Edge_cutout is user-defined

4.2.80 Edge_feature

An Edge_feature is a type of Structural_cutout (see 4.2.264) that is a feature along an edge of a Structural_part. An Edge_feature specifies the removal of material along an entire edge or parts of it for the purpose of weld preparation, for example. Many types of Edge_features may be distinguished. These shall be specified using different Corner_cutout_design_definitions, such as:

- Round_corner_design_definition,
- Inward_round_corner_design_definition,
- Outward_round_corner_design_definition,
- Bevel_design_definition,
- Shear_bevel_design_definition,
- Rectangular_cutback_corner_design_definition.

4.2.81 Edge_feature_functional_definition

An Edge_feature_functional_definition is a type of Functional_definition (see 4.2.112) that defines the function that an Edge_feature may be designed for.

The data associated with an Edge_feature_functional_definition are the following:

- defined_for;
- function.

4.2.81.1 defined_for

The defined_for specifies that the Edge_feature that this function is assigned to. See 4.3.54 for the application assertion.

4.2.81.2 function

The function specifies the functionality of the Edge_feature that this definition points to.

The value of the function shall be one of the following:

- bevel;

- chamfer;
- used defined.

NOTE See 4.2.81.2.1 - 4.2.81.2.3 for the definition of each allowable value for `endcut_shape_type`.

4.2.81.2.1

bevel

the result of removal of material for the purpose of weld preparation

4.2.81.2.2

chamfer

the result of removal of material for the purpose of thickness adoption between plates

4.2.81.2.3

used defined

a function of an `Edge_feature` that is defined by the user

4.2.82 Electrode_chemical_composition

An `Electrode_chemical_composition` specifies the chemical composition of the wire electrode. All chemical components within an electrode wire, except steel, are listed in percentage to give the identified alloy components value.

The data associated with an `Electrode_chemical_composition` are the following:

- aluminum;
- carbon;
- chromium;
- copper;
- manganese;
- molybdenum;
- nickel;
- phosphorus;
- silicon;
- sulphur;
- titanium;

— vanadium;

— zirconium.

4.2.82.1 aluminum

The aluminum specifies the percentage of the chemical component aluminum (Al), e.g., maximum 0.02%.

4.2.82.2 carbon

The carbon specifies the percentage of the chemical component carbon (C), e.g., 0.06% to 0.14%.

4.2.82.3 chromium

The chromium specifies the percentage of the chemical component chromium (Cr), e.g., maximum 0.15%.

4.2.82.4 copper

The copper specifies the percentage of the chemical component copper (Cu), e.g., maximum 0.35%.

4.2.82.5 manganese

The manganese specifies the percentage of the chemical component manganese (Mn), e.g., 0.90% to 1.30%.

4.2.82.6 molybdenum

The molybdenum specifies the percentage of the chemical component molybdenum (Mo), e.g., maximum 0.15%.

4.2.82.7 nickel

The nickel specifies the percentage of the chemical component nickel (Ni), e.g., maximum 0.15%.

4.2.82.8 phosphorus

The phosphorus specifies the percentage of the chemical component phosphorus (P), e.g., maximum 0.025%.

4.2.82.9 silicon

The silicon specifies the percentage of the chemical component silicon (Si), e.g., 0.50% to 0.80%.

4.2.82.10 sulphur

The sulphur specifies the percentage of the chemical component sulphur (S), e.g., maximum 0.025%.

4.2.82.11 titanium

The titanium specifies the percentage of the chemical component titanium (Ti), e.g., maximum 0.15%.

4.2.82.12 vanadium

The vanadium specifies the percentage of the chemical component vanadium (V), e.g., maximum 0.03%.

4.2.82.13 zirconium

The zirconium specifies the percentage of the chemical component zirconium (Zr), e.g., maximum 0.15%.

4.2.83 Elliptical_cutout_design_definition

An Elliptical_cutout is a type of Interior_cutout_design_definition (see 4.2.128) that is given by the formula $(x/a)^2 + (y/b)^2 = 1$, with a and b being the half axis of the opening.

The data associated with an Elliptical_cutout_design_definition are the following:

— half_axis_a;

— half_axis_b.

4.2.83.1 half_axis_a

The half_axis_a specifies the along x-axis.

4.2.83.2 half_axis_b

The half_axis_b specifies the along y-axis.

See figure 18 for Elliptical_cutout.

4.2.84 Elongated_oval_cutout_design_definition

An Elongated_oval_cutout is a type of Interior_cutout_design_definition (see 4.2.128) that is an opening either egg-shaped or shaped like a slot.

The data associated with an `Elongated_oval_cutout_design_definition` are the following:

- `distance`;
- `r1`;
- `r2`;
- `r3`;
- `r3_x`;
- `r3_y`.

4.2.84.1 distance

The `distance` specifies the distance from circle centre with radius `r1` (=origin) to circle centre with radius `r2` in direction of pos. x-axis.

4.2.84.2 r1

The `r1` specifies the radius of circle arc with centre at the origin.

4.2.84.3 r2

The `r2` specifies the radius of circle arc to the right (along x-axis) of circle given by `r1`.

4.2.84.4 r3

The `r3` specifies the radius of the "sides" of the egg shaped opening. The `r3` need not to be specified for a particular `Elongated_oval_cutout_design_definition`.

4.2.84.5 r3_x

The `r3_x` specifies the offset in x-direction of centre of circle with radius `r2`. The `r3_x` need not to be specified for a particular `Elongated_oval_cutout_design_definition`.

4.2.84.6 r3_y

The `r3_y` the offset in y-direction of centre of circle with radius `r2`. The `r3_y` need not to be specified for a particular `Elongated_oval_cutout_design_definition`.

See figure 18 for `Elongated_oval_cutout`.

4.2.85 Envisaged_version_creation

An `Envisaged_version_creation` is a type of `Versionable_object_change_event` (see 4.2.308) that is the event leading to a new versionable object. The event is an envisaged event and has not yet happened. The Definition (see 4.2.57, `Item_structure` (see 4.2.134) or `Item_relationship` (see 4.2.133) as the subject of the event does not yet exist and is described in terms of descriptive, non-formal properties.

The data associated with an `Envisaged_version_creation` are the following:

- `base`;
- `category`.

4.2.85.1 base

The `base` specifies the `Versionable_objects` the envisaged new version is derived from. See 4.3.55 for the application assertion.

4.2.85.2 category

The `category` specifies the category the envisaged versionable object belongs to.

4.2.86 Event

An `Event` identifies that something has happened at a certain time, activated by a certain person for a certain reason.

The data associated with an `Event` are the following:

- `caused_by`;
- `caused_when`;
- `description`.

4.2.86.1 caused_by

The `caused_by` specifies the person creating the `Event`. See 4.3.57 for the application assertion.

4.2.86.2 caused_when

The `caused_when` specifies the date and time, the `Event` occurred. See 4.3.56 for the application assertion.

4.2.86.3 description

The description specifies the textural description of significant features, reason of the Event.

4.2.87 Explicit_feature_design_definition

An `Explicit_feature_design_definition` is a type of `Feature_design_definition` (see 4.2.96) that describes the design of a `Feature` (see 4.2.95) by the explicit representation of its shape. The shape is given as a geometric model without parameters and without a reference to moulded geometry. This `Feature_design_definition` is best suited for the purely visual presentation of the `Feature`. The functionality of the specific `Feature` is not included in this description. The data associated with an `Explicit_feature_design_definition` are the following:

- `representations`.

The `representations` describe the geometric shape of the `Explicit_feature_design_definition`. Only one `Shape_representation` shall be in the set of `representaions`. See 4.3.58 for the application assertion.

4.2.88 Explicit_profile_cross_section

An `Explicit_profile_cross_section` is a type of `Profile_cross_section` (see 4.2.206) that is the explicit geometric definition of the cross-section of a rolled profile.

The data associated with an `Envisaged_version_creation` are the following:

- `cross_section_geometry`;
- `local_coordinate_system`.

4.2.88.1 cross_section_geometry

The `cross_section_geometry` specifies the closed planar curve that defines the outer boundary of the profile. See 4.3.60 for the application assertion.

4.2.88.2 local_coordinate_system

The `local_coordinate_system` specifies the point and pair of axes on the `cross_section_geometry` that that specifies the point and orientation that is used to place the explicit cross-section geometry onto the extrusion trace. See 4.3.59 for the application assertion.

4.2.89 Extended_conversion_based_unit

An `Extended_conversion_based_unit` is a type of `Named_unit` (see 4.2.164) that is defined based on a `Extended_measure_with_unit` (see 4.2.91). It is an extension of the P41 Entity `Conversion_based_unit`.

EXAMPLE 1 An inch is an `Extended_conversion_based_unit`. It is from the Imperial system, its name is 'inch' and it can be related to the `Si_unit`, millimeter, through an `Extended_measure_with_unit` whose value is 25.4 millimeter. A foot is also an `Extended_conversion_based_unit`. It is from the Imperial system, its name is 'foot' and it can be related to the `si_unit`, millimeter, either directly or through the unit called 'inch'.

The data associated with an `Extended_conversion_based_unit` are the following:

- `conversion_factor`;
- `name`.

4.2.89.1 `conversion_factor`

The `conversion_factor` specifies the physical quantity from which the `Extended_conversion_based_unit` is derived. See 4.3.61 for the application assertion.

4.2.89.2 `name`

The `name` specifies the word or group of words by which the `Extended_conversion_based_unit` is referred to.

4.2.90 `Extended_global_uncertainty_assigned_context`

An `Extended_global_uncertainty_assigned_context` is a type of `representation_context` (see ISO 10303 Part 43) that is a `representation_context` that specifies uncertainty for the elements of representation that are associated with it. The uncertainty is specified by instances of `extended_uncertainty_measure_with_unit`, and applies to all elements of representation that are expressed in the same measure and that are associated with a representation that has this `extended_global_uncertainty_assigned_context`. Compared with the `global_uncertainty_assigned_context` from Part 43 this one references a set of `extended_uncertainty_measure_with_units`, not a set of `uncertainty_measure_with_units` as in Part 43.

NOTE The precedence rules that apply to uncertainties in numeric quantities are specified in 4.2.6 in Part 43.

EXAMPLE 1 An instance of `Extended_global_uncertainty_assigned_context` specifies uncertainty of 0.01m with respect to lengths. Unless modified by the precedence rules specified in 4.2.6 in Part 43, this uncertainty applies to each length that occurs in each `representation_item` associated with a representation that has this `Extended_global_uncertainty_assigned_context`.

The data associated with an `Extended_global_uncertainty_assigned_context` are the following:

- `uncertainty`.

The `uncertainty` specifies the instances of `extended_uncertainty_measure_with_unit` that apply in the `representation_context`. See 4.3.62 for the application assertion.

4.2.91 Extended_measure_with_unit

An `Extended_measure_with_unit` is the specification of a physical quantity as defined in ISO 31 (clause 2).

The data associated with an `Extended_measure_with_unit` are the following:

- `unit_component`;
- `value_component`.

4.2.91.1 unit_component

The `unit_component` specifies the unit of the physical quantity.

4.2.91.2 value_component

The `value_component` specifies the value of the physical quantity when expressed in the specified units.

4.2.92 Extended_uncertainty_measure_with_unit

An `Extended_uncertainty_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) that is an `extended_measure_with_unit` that specifies the uncertainty that applies to a type of measure. An `extended_uncertainty_measure_with_unit` applies to every `representation_item` that uses the type of measure specified in the `value_component` of the `extended_uncertainty_measure_with_unit`. The difference to the `uncertainty_measure_with_unit` in Part 43 is that this entity type here is a subtype of `extended_measure_with_unit` and not of `measure_with_unit`.

The data associated with an `Extended_uncertainty_measure_with_unit` are the following:

- `description`;
- `name`.

4.2.92.1 description

The `description` specifies the text that characterizes the `extended_uncertainty_measure_with_unit`. The value of the `description` need not be specified.

4.2.92.2 name

The `name` specifies the label by which the `extended_uncertainty_measure_with_unit` is known.

4.2.93 External_instance_reference

An `External_instance_reference` represents an instance of an entity that does not exist in the same scope.

NOTE The entity whose instance is represented must be a subtype of either `Definable_object` or `Definition` in order to be referable via a global unique identifier. If the type of the externally referenced instance is also a `Versionable_object` and a specific version shall be referred to The version shall be used as well.

The data associated with an `External_instance_reference` are the following:

- `entity_type`;
- `schema_name`;
- `target_guid`;
- `version`.

4.2.93.1 entity_type

The `entity_type` specifies the name of the type of the externally referenced instance.

4.2.93.2 schema_name

The `schema_name` specifies the name of the schema the type of the externally referenced instance is defined in.

4.2.93.3 target_guid

The `target_guid` specifies the global unique identifier of the externally referenced instance. See 4.3.63 for the application assertion.

4.2.93.4 version

The `version` specifies the version of the externally defined instance. The version need not to be specified for a particular `External_instance_reference`.

4.2.94 External_reference

An `External_reference` is the abstract denotation of a data source external to the data set where an instance of this entity exists.

EXAMPLE 1 a WWW uniform resource locator denotes such a data source

The data associated with an `External_reference` are the following:

- description;
- location.

4.2.94.1 description

The description specifies the some additional information.

4.2.94.2 location

The location specifies the location of an external reference, in case of an `Universal_resource_locator` a computer accessible by a specified transmission protocol.

4.2.95 Feature

A Feature is a type of Item (see 4.2.132). Features are design and production details of the structural elements of a ship, such as corner-features, edge-features or interior-features (openings).

NOTE A Feature represents just the function or idea of a feature. Its details may be given by adding Definition subtypes, such as `Explicit_feature_design_definition` (see 4.2.87) that reference this Feature.

NOTE No provision is made for a Feature on this level to reference other Items, e.g. an owner of the feature.

4.2.96 Feature_design_definition

A `Feature_design_definition` is a type of `Design_definition` (see 4.2.63) that describes the definition of a Feature (see 4.2.95) for the life-cycle phase of design.

The data associated with a `Feature_design_definition` are the following:

- `defined_for`.

The `defined_for` specifies the Feature that the `Feature_design_definition` is defined for.

4.2.97 Feature_relationship

A `Feature_relationship` is a type of `Item_relationship` (see 4.2.133) that defines the association of a Feature (see 4.2.95) with another Item (see 4.2.132).

EXAMPLE 1 `Feature_relationship` may be applied to indicate that two Seams are parallel or that one Seam is parallel to a plate boundary. `Feature_relationship` does not replace the parent attribute of `Structural_feature`.

The data associated with a `Feature_relationship` are the following:

— item_1.

The item_1 specifies the relating item of the relationship shall be a Feature. See 4.3.65 for the application assertion.

4.2.98 Fillet_weld

A Fillet_weld is a type of Welded_joint_design_definition (see 4.2.324) that is a weld where the weld bead is laid down along the outside of the part. The degree of penetration achieved by this type of weld is dependent on the thickness and arrangement of the parts being welded and usually, is less than that attainable with a groove weld. The advantage it offers in elimination of edge preparation, makes this a highly desirable weld detail.

The data associated with a Fillet_weld are the following:

- endcut_shape_type;
- sidedness.

4.2.98.1 endcut_shape_type

The endcut_shape_type specifies the indicator as to explain the endcut shape of parts being welded at a fillet weld joint.

The value of the endcut_shape_type shall be one of the following:

- fillet J;
- fillet V;
- used defined.

NOTE See 4.2.98.1.1 - 4.2.98.1.2 for the definition of each allowable value for endcut_shape_type.

4.2.98.1.1

fillet_J

particular shape of the edge curved cut form of plate or end cut form of profile parts being welded by fillet weld

4.2.98.1.2

fillet_V

particular shape of the edge plane cut form of plate or end cut form of profile parts being welded by fillet weld

4.2.98.1.3

used_defined

other fillet cut form defined by the user

4.2.98.2 sidedness

The sidedness specifies the identifier as to whether the fillet weld is to be made at one sides or at both sides of the welded joint.

The value of the sidedness shall be one of the following:

- both sides;
- one side.

NOTE See 4.2.98.2.1 - 4.2.98.2.2 for the definition of each allowable value for sidedness.

4.2.98.2.1**both sides**

both sides of a weld joint are welded

4.2.98.2.2**one side**

one side of a weld joint is welded

4.2.99 Flanged_plate_cross_section

A Flanged_Plate_cross_section is a type of Proprietary_profile_cross_section (see 4.2.217) commonly used in shipbuilding in place of an Angle cross section. Flat plate is knicked into an angular shape.

The data associated with a Flanged_plate_cross_section are the following:

- depth;
- radius;
- thk;
- width.

4.2.99.1 depth

The width specifies the depth of the web.

4.2.99.2 radius

The width specifies the radius between the flange and the web.

4.2.99.3 `thk`

The width specifies the thickness of the plate from which the cross-section is manufactured.

4.2.99.4 `width`

The width specifies the width of the flange.

4.2.100 `Flanged_profile_cross_section`

A `Flanged_profile_cross_section` is a type of `Parametric_profile_cross_section` (see 4.2.178). A `Flanged_profile_cross_section` may be a `Flanged_Plate_cross_section`. A `Flanged_profile_cross_section` is a generic type of standard shape that can be distinguished as having an identifiable flange and web.

4.2.101 `Flare_area`

A `Flare_area` describes properties of the fore-ship that indicate the danger for buckling in the fore-ship due to upwards pushing sea.

NOTE Flare area may be a critical aspect especially for slim ships that go at high speeds.

The data associated with a `Flare_area` are the following:

- `area`;
- `height`.

4.2.101.1 `area`

The `area` specifies the difference between the area of the weather deck forward of 0.2 of the rule length of the ship measured from forward perpendicular and the corresponding area at the summer-waterline.

4.2.101.2 `height`

The `height` specifies the distance from the baseline of the ship to the weather deck measured at forward perpendicular and on the ship centre line.

4.2.102 `Flat_bar_cross_section`

A `Flat_bar_cross_section` is a type of `Bar_profile_cross_section` (see 4.2.15) identified by a solid rectangular shape.

NOTE ISO 1035-3 specifies dimensions for some Flat bars.

The data associated with a `Flat_bar_cross_section` are the following (see Figure 20)

- `depth`;
- `width`.

4.2.102.1 depth

The `depth` specifies the dimension of the longer of the two sides, if unequal.

4.2.102.2 width

The `width` specifies the dimension of the shorter of the two sides, if unequal.

4.2.103 Flow_rate_measure_with_unit

A `Flow_rate_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is the flow rate of a medium.

4.2.104 Flow_rate_unit

A `Flow_rate_unit` is a type of `Derived_unit` (see 4.2.61) that is the unit of the volume flow of a medium within a time period, that is volume per time.

4.2.105 Force_measure_with_unit

A `Force_measure_with_unit` is a type of `Extended_measure_with_unit` where the physical quantity is a force.

4.2.106 Force_unit

A `Force_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which force is measured.

4.2.107 Frame_table

A `Frame_table` is a type of `Longitudinal_table` (see 4.2.151) that is a spacing table whose positions are a reference for the location of frames and are located on the global x axis.

NOTE Frames are used for the internal structure of the ship and they are structural elements. A ship can have more than 100 frames. The intersection curve between a frame and the hull moulded form is a curve of transversal section through the ship hull.

The data associated with a `Frame_table` are the following:

- `spacing_table_representations`.

The `spacing_table_representations` specifies the longitudinal positions of the frames.

4.2.108 Free_form_interior_cutout_design_definition

A `Free_form_interior_cutout_design_definition` is a type of `Interior_cutout_design_definition` (see 4.2.128) that specifies an opening that can not be represented by the basic openings. A bounding (closed) curve is defined that represents the shape of the opening.

The data associated with a `Free_form_interior_cutout_design_definition` are the following:

- `bounding_curve`.

The `bounding_curve` specifies the closed curve representing the opening. See 4.3.66 for the application assertion.

4.2.109 Freeboard_characteristics

A `Freeboard_characteristics` is a type of `General_characteristics_definition` (see 4.2.113) that give details of the aggriment of freeboard of a ship.

The data associated with a `Freeboard_characteristics` are the following:

- `applicable_loadline`;
- `date_freeboard_assigned`;
- `freeboard`;
- `freeboard_assigned_by`.

4.2.109.1 applicable_loadline

The `applicable_loadline` is the loadline which can be applied. See 4.3.68 for the application assertion.

4.2.109.2 date_freeboard_assigned

The `date_freeboard_assigned` is the date and time the freeboard is assigned. See 4.3.67 for the application assertion.

4.2.109.3 freeboard

The `freeboard` is the assigned freeboard and is the difference between the actual depth at side including the thickness of the stringer plate and wooden deck sheathing, if fitted, and the summer load draught.

4.2.109.4 freeboard_assigned_by

The `freeboard_assigned_by` is the organization which made the freeboard assignment. See 4.3.69 for the application assertion.

4.2.110 Frequency_measure_with_unit

A `Frequency_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is a frequency.

4.2.111 Frequency_unit

A `Frequency_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which occurrences per time unit are measured.

4.2.112 Functional_definition

A `Functional_definition` is a type of `Definition` (see 4.2.57) that is the supertype for all kinds of functional definitions.

The data associated with a `Functional_definition` are the following:

- `local_units`;
- `user_def_function`.

4.2.112.1 local_units

The `local_units` specifies that a `Functional_definition` shall not define local units.

4.2.112.2 user_def_function

The `user_def_function` specifies the Each non-abstract SUBTYPE of `Functional_definition` should declare (or inherit) an attribute function with a special ENUMERATION type containing the possible functions for the `Definable_object` subtype it points to with at least one entry called `USER_DEFINED`. In the case of `function = USER_DEFINED` the `user_def_function` attribute has to be used to determine the user defined function. The `user_def_function` need not to be specified for a particular `Functional_definition`.

4.2.113 General_characteristics_definition

A `General_characteristics_definition` is a type of `Definition` (see 4.2.57) that provides a major part of the documentation of the vessel. It includes primary dimensions and capacities due to the contract of the product (ship).

The data associated with a `General_characteristics_definition` are the following:

— `defined_for`.

The `defined_for` specifies a Ship or set of Ships for which the `General_characteristics_definition` applies. See 4.3.70 for the application assertion.

4.2.114 Global_axis_placement

A `Global_axis_placement` is a type of `General_characteristics_definition` (see 4.2.113) that defines a fixed system of right handed orthogonal axes to which geometric data are referred. A `Global_axis_placement` shall have a: - positive z axis in an upwards direction starting from the base of the ship, - positive x axis running along the ship on the intersection of the centreline with the base and is in one case directed from the after part of the ship to the forward part of the ship or in the other case is directed from the forward part of the ship to the aft part of the ship, - origin of the global axis placement can be any point on the x axis. The distance of the after perpendicular from the origin and the orientation of the x axis shall be specified. If any other system of axes is used, local or global, then the transformation relations between it and the `Global_axis_placement` shall be specified, see Figure 21.

The data associated with a `Global_axis_placement` are the following:

- `after_perpendicular_offset`;
- `orientation`.

4.2.114.1 after_perpendicular_offset

The `after_perpendicular_offset` specifies the distance from the origin of the `Global_axis_placement` to the after perpendicular.

4.2.114.2 orientation

The `orientation` specifies the direction of the x-axis.

The value of `orientation` shall be one of the following:

- `aft pointing`;
- `forward pointing`.

NOTE See 4.2.114.2.1 - 4.2.114.2.2 for the definition of each allowable value for `orientation`.

`aft pointing`: an orientation of a right handed ship co-ordinate system that has the positive x axis from the forward part of the ship directed to the aft part of the ship.

`forward pointing`: an orientation of a right handed ship co-ordinate system that has the positive x axis from the aft part of the ship directed to the forward part of the ship.

4.2.115 Groove_weld

A Groove_weld is a type of Welded_joint_design_definition (see 4.2.324) which is a weld between the structural parts where, either due to increased part thickness or highly stressed joints, deeper penetration of weld materials is required. In order to achieve this penetration, the edges of the parts to be joined require edge preparation such as bevel/chamfer features or grinding features. This type of weld is usually stronger but more costly than a fillet weld. It is usually used in highly stressed and critical areas of the ship, such as the shell plating, decks, bulkheads, etc.

An example for the groove weld seen in Figure 22.

The data associated with a Groove_weld are the following:

- sidedness;
- weld_backing_type;
- weld_joint_spacer.

4.2.115.1 sidedness

The sidedness specifies the identifier as to denote whether the groove weld is to be made at one sides or at both sides of the welded joint.

The value of the sidedness shall be one of the following:

- both sides.
- one side;

NOTE See 4.2.115.1.1 - 4.2.115.1.2 for the definition of each allowable value for sidedness.

4.2.115.1.1

both sides

both sides of a weld joint are welded

4.2.115.1.2

one side

one side of a weld joint is welded

4.2.115.2 weld_backing_type

The weld_backing_type specifies the type of required underside (back) of the weld in order to prevent "blow through" and/or eliminate the need for back gouging. Proper selection of backing reduces manufacturing costs by making it possible to joint parts by welding from a single side.

EXAMPLE 1 A cut-down channel bar used as a deck beam, the upper flange providing the backing bar for a deck panel butt weld, made by machine above.

The weld_backing_type need not to be specified for a particular Groove_weld.

The value of the weld_backing_type shall be one of the following:

- ceramic tape;
- copper backing bar;
- flare backing ring;
- none;
- permanent;
- permanent backing ring;
- removable backing ring;
- user defined.

NOTE See 4.2.115.2.1 - 4.2.115.2.8 for the definition of each allowable value for weld_backing-type.

4.2.115.2.1

ceramic tape

a backing tape made of ceramic

4.2.115.2.2

copper backing bar

a backing bar made of copper

4.2.115.2.3

flare backing ring

a backing ring of flare form

4.2.115.2.4

none

no backing is defined

4.2.115.2.5

permanent

a backing attached at the welded joint permanently

4.2.115.2.6

permanent backing ring

a backing ring attached at the welded joint permanently

4.2.115.2.7**removable backing ring**

a removable backing ring

4.2.115.2.8

user defined a backing defined by the user

4.2.115.3 weld_joint_spacer

The weld_joint_spacer specifies the an identifier as to whether or not the joint requires the use of additional material to maintain the weld dimensions before or during the welding process. This is purely a boolean identifier that indicates that a spacer is required. the geometric definition of the spacer must be derived from the joint detailed dimensions.

4.2.116 Heat_rate_measure_with_unit

A Heat_rate_measure_with_unit is antype of Extended_measure_with_unit (see 4.2.91) where the physical quantity is an energy flow.

4.2.117 Heat_rate_unit

A Heat_rate_unit is a type of Derived_unit (see 4.2.61) that is the unit of the physical quantity of an energy flow.

4.2.118 Homogeneous_ship_material_property

A Homogeneous_ship_material_property is a type of Ship_material_property (see 4.2.242) that is used to specify the raw material of this class by its physical properties.

The data associated with a Homogeneous_ship_material_property are the following:

- defined_for;
- poisson_ratio;
- stress_of_fracture;
- thermal_expansion_coefficient;
- yield_point;
- youngs_module.

4.2.118.1 defined_for

The `defined_for` redeclare the `defined_for` attribute in order to allow the `Homogeneous_ship_material_property`, as a type of definition, to point to `Parts` only. See 4.3.71 for the application assertion.

4.2.118.2 poisson_ratio

The `poisson_ratio` is the negative ratio between longitudinal and transversal strain.

4.2.118.3 stress_of_fracture

The `stress_of_fracture` specifies the crossing the `stress_of_fracture` causes the end of the materially coherence.

4.2.118.4 thermal_expansion_coefficient

The `thermal_expansion_coefficient` specifies the dilatation in meters caused by a temperature rise of 1 K.

4.2.118.5 yield_point

The `yield_point` specifies the crossing the `yield_point` causes plastic deformations.

4.2.118.6 youngs_module

The `youngs_module` specifies the youngs module is the ratio between stress and strain while the material behaves linear (Hookes law)

NOTE The sheer module G is derivable from $G = E / (2 * (1 + my))$ with the youngs module E and the poisson ratio my .

4.2.119 Hull_cross_section

A `Hull_cross_section` is a type of both `Item` (see 4.2.132) and `Item_structure` (see 4.2.134) that is a collection of all those parts of the structure of a ship which are relevant to its longitudinal strength.

NOTE For this the ship is viewed as a beam, with a length of approximately the length between perpendiculars and with cross sectional properties varying over its length. A `Hull_cross_section` includes all data that are necessary for its design and its approval with respect to class rule requirements.

EXAMPLE 1 Panels in bottom, side, deck, bulkheads, girders, and stringers are examples of `Structural_systems` that may be relevant in this context.

The model is applicable to any kind of ship.

A `Hull_cross_section` is an Item in the context of a Ship. It shall have a location along the global longitudinal axis with transverse and vertical axis aligned with the corresponding ones in the global hull co-ordinate system. The location shall be specified by a `Local_co_ordinate_system_with_station_reference` definition. A `Hull_cross_section` is also an Item_structure that references Compartments and Structural_systems. These items shall have Definitions that specify the information that is necessary for the hull cross section approval by classification society; these definitions are `Class_compartment_requirement_definition` and `Class_deck_load_requirement_definition`. All the definitions that together describe a `Hull_cross_section` are collected by `Revision_with_context` with `Revision_with_context.members` being these definitions, one of which is a `Hull_cross_section_design_definition`, and `Revision_with_context.context_of_revision` being the `Hull_cross_section`.

Figure 23 shows a hull cross section of a cargo ship.

4.2.120 `Hull_cross_section_approval`

A `Hull_cross_section_approval` is a type of `Hull_cross_section_result` (see 4.2.123) that confirms the approval of a `Hull_cross_section_design_definition` (see 4.2.121).

NOTE A `Hull_cross_section_design_definition` is approved if all its constituents are approved and if the requirements as stated in `Hull_cross_section_result` are fulfilled.

4.2.121 `Hull_cross_section_design_definition`

A `Hull_cross_section_design_definition` is a type of `Design_definition` (see 4.2.63) that describes the definition of a `Hull_cross_section` (see 4.2.119) from the view of design for classification approval.

The data associated with a `Hull_cross_section_design_definition` are the following:

- `defined_for`;
- `flare_area_buckling`;
- `symmetry`.

4.2.121.1 `defined_for`

The `defined_for` specifies the one `Hull_cross_section` item that a `Hull_cross_section_design_definition` is defined for. See 4.3.73 for the application assertion.

NOTE A `Hull_cross_section` may have many `Hull_cross_section_design_definitions`, each describing a different version.

4.2.121.2 flare_area_buckling

The flare_area_buckling specifies the part of the fore-ship that shall withstand the forces that are introduced by upwards pushing sea. See 4.3.72 for the application assertion.

4.2.121.3 symmetry

The symmetry specifies the indication whether the hull cross section is symmetrical to the mid-ship plane.

NOTE Nonetheless the whole hull cross section will be provided.

4.2.122 Hull_cross_section_rejection

A Hull_cross_section_rejection is a type of Hull_cross_section_result (see 4.2.123) that confirms that a Hull_cross_section_design_definition (see 4.2.121) is not approved.

4.2.123 Hull_cross_section_result

A Hull_cross_section_result specifies is a type of Structural_class_approval_result (see 4.2.261) that is the result of the assessment of a Hull_cross_section_design_definition (see 4.2.121) against for example the rules of a classification society. Subtypes of this entity indicate whether the item passed the verification or not. If the Hull_cross_section_design_definition contains any item that is not approved, the whole Hull_cross_section (see 4.2.119) is disapproved. In addition there is a set of requirements that the Hull_cross_section_design_definition as a whole shall fulfill to be approved. These are specified by this entity.

The data associated with a Hull_cross_section_result are the following:

- actual_first_moment_horizontal;
- actual_first_moment_vertical;
- actual_inertia_cross;
- actual_inertia_horizontal;
- actual_inertia_torsion;
- actual_inertia_vertical;
- actual_neutral_axis_position_horizontal;
- actual_neutral_axis_position_vertical;
- actual_section_area;

- actual_section_modulus_bottom;
- actual_section_modulus_coaming;
- actual_section_modulus_deck;
- actual_stress_bottom;
- actual_stress_coaming;
- actual_stress_deck;
- actual_twist_centre_horizontal;
- actual_twist_centre_vertical;
- direction_largest_inertia;
- direction_least_inertia;
- max_moment;
- max_shear_force;
- meter_weight;
- min_moment;
- min_shear_force;
- required_section_modulus_bottom;
- required_section_modulus_coaming;
- required_section_modulus_deck;
- required_stress_bottom;
- required_stress_coaming;
- required_stress_deck;
- subject.

4.2.123.1 actual_first_moment_horizontal

The `actual_first_moment_horizontal` specifies the calculated first moment about the horizontal neutral axis.

4.2.123.2 `actual_first_moment_vertical`

The `actual_first_moment_vertical` specifies the calculated first moment about the vertical neutral axis.

4.2.123.3 `actual_inertia_cross`

The `actual_inertia_cross` specifies the calculated cross moment of inertia.

4.2.123.4 `actual_inertia_horizontal`

The `actual_inertia_horizontal` specifies the calculated moment of inertia about the horizontal neutral axis.

4.2.123.5 `actual_inertia_torsion`

The `actual_inertia_torsion` specifies the calculated torsion moment of inertia.

4.2.123.6 `actual_inertia_vertical`

The `actual_inertia_vertical` specifies the calculated moment of inertia about the vertical neutral axis.

4.2.123.7 `actual_neutral_axis_position_horizontal`

The `actual_neutral_axis_position_horizontal` specifies the horizontal distance in this cross section between the longitudinal axis of the global co-ordinate system and the neutral axis.

4.2.123.8 `actual_neutral_axis_position_vertical`

The `actual_neutral_axis_position_vertical` specifies the vertical distance in this cross section between the longitudinal axis of the global co-ordinate system and the neutral axis.

4.2.123.9 `actual_section_area`

The `actual_section_area` specifies the calculated area of the `hull_cross_section_design_definition` as determined by the analysis.

4.2.123.10 `actual_section_modulus_bottom`

The `actual_section_modulus_bottom` specifies the actual section modulus for the bottom as a result of the proposed structure.

4.2.123.11 actual_section_modulus_coaming

The `actual_section_modulus_coaming` specifies the actual section modulus for the coaming as a result of the proposed structure.

4.2.123.12 actual_section_modulus_deck

The `actual_section_modulus_deck` specifies the actual section modulus for the deck as a result of the proposed structure.

4.2.123.13 actual_stress_bottom

The `actual_stress_bottom` specifies the calculated stress in the bottom.

4.2.123.14 actual_stress_coaming

The `actual_stress_coaming` specifies the calculated stress in the coaming.

4.2.123.15 actual_stress_deck

The `actual_stress_deck` specifies the calculated stress in the weather deck.

4.2.123.16 actual_twist_centre_horizontal

The `actual_twist_centre_horizontal` specifies the horizontal distance in this cross section between the longitudinal axis of the global co-ordinate system and the centre of twisting.

4.2.123.17 actual_twist_centre_vertical

The `actual_twist_centre_vertical` specifies the vertical distance in this cross section between the longitudinal axis of the global co-ordinate system and the centre of twisting.

4.2.123.18 direction_largest_inertia

The `direction_largest_inertia` specifies the calculated direction angle of the largest moment of inertia.

4.2.123.19 direction_least_inertia

The `direction_least_inertia` specifies the calculated direction angle of the least moment of inertia.

4.2.123.20 max_moment

The `max_moment` specifies the required maximum still water bending moment.

4.2.123.21 max_shear_force

The max_shear_force specifies the required maximum shear force.

4.2.123.22 meter_weight

The meter_weight specifies the weight of longitudinal structural members per meter.

4.2.123.23 min_moment

The min_moment specifies the required minimum still water bending moment.

4.2.123.24 min_shear_force

The min_shear_force specifies the required minimum shear force.

4.2.123.25 required_section_modulus_bottom

The required_section_modulus_bottom specifies the required section modulus for the bottom.

4.2.123.26 required_section_modulus_coaming

The required_section_modulus_coaming specifies the required section modulus for the coaming.

4.2.123.27 required_section_modulus_deck

The required_section_modulus_deck specifies the required section modulus for the deck.

4.2.123.28 required_stress_bottom

The required_stress_bottom specifies the required stress in the bottom.

4.2.123.29 required_stress_coaming

The required_stress_coaming specifies the required stress in the coaming.

4.2.123.30 required_stress_deck

The required_stress_deck specifies the required stress in the weather deck.

4.2.123.31 subject

The subject specifies the approved/rejected Hull_cross_section. See 4.3.74 for the application assertion.

4.2.124 Inertia_moment_measure_with_unit

An `Inertia_moment_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is the area moment of inertia.

4.2.125 Inertia_moment_unit

An `Inertia_moment_unit` is a type of `Derived_unit` (see 4.2.61) that is the unit of a shape parameter that indicates the ability of a cross section to withstand bending. This area moment of inertia is measured relative to an axis and is the sum of the products of all area elements within a plane area with their squared vertical distances from the axis.

4.2.126 Integer_property_value

An `Integer_property_value` is a type of `Property_value` (see 4.2.216) that provides a value for a property of type `INTEGER`.

The data associated with an `Integer_property_value` are the following:

— `val`.

The `val` specifies the value of type `INTEGER`.

4.2.127 Interior_cutout

An `Interior_cutout` is a type of `Structural_cutout` (see 4.2.264) that specifies holes in structural parts, such as manholes, access-holes, lightening-holes. Many types of `Interior_cutouts` may be distinguished. These shall be specified using different interior cutout design definitions, such as:

- `Free_form_interior_cutout_design_definition`,
- `Circular_cutout_design_definition`,
- `Elliptical_cutout_design_definition`,
- `Elongated_oval_cutout_design_definition`,
- `Rectangular_cutout_design_definition`,
- `Round_corner_rectangular_cutout_design_definition`,
- `Round_edge_rectangular_cutout_design_definition`,
- `Triangular_cutout_design_definition`.

4.2.128 Interior_cutout_design_definition

An Interior_cutout_design_definition is a type of Feature_design_definition (see 4.2.96) that specifies a design definition for an opening.

The data associated with an Interior_cutout_design_definition are the following:

- defined_for;

4.2.128.1 defined_for

The defined_for specifies the Interior_cutouts that a Interior_cutout_design_definition shall be valid for. See 4.3.75 for the application assertion.

4.2.129 Interior_cutout_functional_definition

An Interior_cutout_functional_definition is a type of Functional_definition (see 4.2.112) that defines the function that an Interior_cutout (see 4.2.127) may be designed for.

The data associated with an Interior_cutout_functional_definition are the following:

- defined_for;
- function.

4.2.129.1 defined_for

The defined_for specifies the Interior_cutout feature that this function is assigned to.

4.2.129.2 function

The function specifies the functionality of the Interior_cutout that this definition points to. See 4.3.76 for the application assertion.

The value of the function shall be one of the following:

- access hole;
- air escape;
- air liquid escape;
- foothold;
- lightening hole;

- liquid escape;
- penetration;
- user defined.

NOTE See 4.2.129.2.1 - 4.2.129.2.8 for the definition of each allowable value for function.

4.2.129.2.1

access hole

the Interior_cutout is used as an access hole

4.2.129.2.2

air escape

the Interior_cutout is used as an air escape

4.2.129.2.3

air liquid escape

the Interior_cutout is used as an air and liquid escape

4.2.129.2.4

foothold

the Interior_cutout is used as a foothold

4.2.129.2.5

lightening hole

the Interior_cutout is used as a lightening hole

4.2.129.2.6

liquid escape

the Interior_cutout is used as a liquid escape

4.2.129.2.7

penetration

the Interior_cutout is used as a penetration

4.2.129.2.8

user defined

the function code for the Interior_cutout is user-defined

4.2.130 Intermittent_fillet_weld

An Intermittent_fillet_weld is a type of Fillet_weld (see 4.2.98) where the weld bead is deposited in a sequence of short continuous length, spaced at regular intervals. The combination of a weld length and spacing forms a pattern that is repeated along the length of the weld. The primary use of intermittent welds is to reduce distortion in thinner structural parts and to reduce manufacturing costs in lightly stressed members.

The data associated with an `Intermittent_fillet_weld` are the following:

- `cutout_rules`;
- `end_rules`;
- `fillet_alignment`;
- `fillet_weld_length`;
- `fillet_weld_spacing`;
- `penetration_rules`.

4.2.130.1 cutout_rules

The `cutout_rules` specifies the aspect of an `Intermittent_fillet_weld` that defines the rule of length of weld that must be made on each side of a cutout.

The value of the `cutout_rules` shall be one of the following:

- `none`;
- `fixed rule`;
- `member depth`;
- `percent length`.

NOTE See 4.2.130.1.1 - 4.2.130.1.4 for the definition of each allowable value for `cutout_rules`.

4.2.130.1.1

none

no intermittent weld rule is defined

4.2.130.1.2

fixed rule

fixed intermittent weld rule is defined

4.2.130.1.3

member depth

intermittent weld rule depending on the welded metals

4.2.130.1.4

percent length

intermittent weld rule which is expressed as a percent of the joint length of the base metal being welded

4.2.130.2 end_rules

The `end_rules` specifies the an aspect of an `Intermittent_fillet_weld` that defines the rule of length of weld that must be made on each end.

The value of the `end_rules` shall be one of the following:

- `none`;
- `fixed rule`;
- `member depth`;
- `percent length`.

NOTE See 4.2.130.2.1 - 4.2.130.2.4 for the definition of each allowable value for `end_rules`.

4.2.130.2.1**none**

no intermittent weld rule is defined

4.2.130.2.2**fixed rule**

fixed intermittent weld rule is defined

4.2.130.2.3**member depth**

intermittent weld rule depending on the welded metals

4.2.130.2.4**percent length**

intermittent weld rule which is expressed as a percent of the joint length of the base metal being welded

4.2.130.3 fillet_alignment

The `fillet_alignment` specifies the an indicator as to whether or not each one of paired single fillet weld making up the double fillet weld are aligned with each other or offset from each other.

The value of the `fillet_alignment` shall be one of the following:

- `staggered`;
- `chained`.

NOTE See 4.2.130.3.1 - 4.2.130.3.2 for the definition of each allowable value for `fillet_alignment`.

4.2.130.3.1

staggered

the intermittent fillet welds with offset from each other

4.2.130.3.2

chained

the intermittent fillet welds aligned with each other

4.2.130.4 fillet_weld_length

The `fillet_weld_length` specifies the the measure of the length of continuous weld bead to be deposited. These lengths are spaced at intervals defined by `fillet_weld_spacing` and are adjusted at the ends of the weld and at the location of cutouts.

4.2.130.5 fillet_weld_spacing

The `fillet_weld_spacing` specifies the the measure of the distance between intermittent fillet weld lengths. The value represents the greatest distance between weld lengths that is not welded.

EXAMPLE 1 If the weld length is 75 cm and the spacing is 150 cm than only 33% ($75/75+150$) of physical weld joint is actually welded.

4.2.130.6 penetration_rules

The `penetration_rules` specifies the an aspect of an `Intermittent_fillet_weld` that defines the rule of penetration of weld that must be made on the intermittent weld.

The value of the `penetration_rules` shall be one of the following:

- none;
- fixed rule;
- member depth;
- percent length.

NOTE See 4.2.130.6.1 - 4.2.130.6.4 for the definition of each allowable value for `penetration_rules`.

4.2.130.6.1

none

no intermittent weld rule is defined

4.2.130.6.2

fixed rule

fixed intermittent weld rule is defined

4.2.130.6.3

member depth

intermittent weld rule depending on the welded metals

4.2.130.6.4

percent length

intermittent weld rule which is expressed as a percent of the joint length of the base metal being welded

4.2.131 Inward_round_corner_design_definition

An `Inward_round_corner_design_definition` is a type of `Round_corner_design_definition` (see 4.2.229) that specifies the radius and location of the centre of the circle arc representing the inward round corner. The centre of the circle arc is placed outside the work item with the local origin located at the corner that shall be removed. The transition between the circle arc (surface) and the adjacent edges is not smooth.

The data associated with an `Inward_round_corner_design_definition` are the following:

- `x_offset`;
- `y_offset`.

4.2.131.1 x_offset

The `x_offset` specifies the distance of the centre of the circle arc in positive x-direction.

4.2.131.2 y_offset

The `y_offset` specifies the distance of the centre of the circle arc in positive y-direction.

Figure 24 shows `Inward_round_corner_cutout` as well as `Outward_round_corner_cutout`.

4.2.132 Item

An `Item` is a type of `Definable_object` (see 4.2.56) that is discrete, identifiable thing used in one or more production activities. An `Item` is something (to be) created by a physical or mental activity or (automatically) derived from one or more other `Items`. An `Item` needs not to represent a physically realizable thing. It may also represent some abstract concept like "activity", "task" etc. `Item` provides the functionality to have relationships to other `Items` and to be member in an `Item_structure`. Each entity that is intended to have such a functionality should inherit from `Item`.

The data associated with an `Item` are the following:

- description;
- documentation;
- name;
- ship_context.

4.2.132.1 description

The description specifies the description for SELF. The ship_context need not to be specified for a particular .

4.2.132.2 documentation

The documentation specifies the documentation available (if it exists) for SELF. See 4.3.77 for the application assertion.

4.2.132.3 name

The name specifies the (human readable) name of the concept that is represented by SELF.

4.2.132.4 ship_context

The ship_context specifies the context of SELF in terms of its applicability to a ship. The ship_context need not to be specified for a particular Item. See 4.3.78 for the application assertion.

4.2.133 Item_relationship

An Item_relationship is a type of both Definable_object (see 4.2.56) and Versionable_object (see 4.2.307) that defines the association of two Items (see 4.2.132). This means the related Items share any thing or activity or are dependent from each other. Each entity that is intended to relate Items in any way should inherit from Item_relationship.

The data associated with an Item_relationship are the following:

- ctxt;
- external_item_1;
- external_item_2;
- item_1;
- item_2.

4.2.133.1 ctxtxt

The ctxtxt specifies the describes what is significant about the Items in their relationship in order to identify the impact of changes to a Item on the related Item. The ctxtxt need not to be specified for a particular .

4.2.133.2 external_item_1

The external_item_1 specifies the relating item of the relationship, in case it is external to the instance model of itself. See 4.3.79 for the application assertion.

4.2.133.3 external_item_2

The external_item_2 specifies the related item of the relationship, in case it is external to the instance model of itself. See 4.3.79 for the application assertion.

4.2.133.4 item_1

The item_1 specifies the relating item of the relationship, in case it is in the same instance model as itself. See 4.3.80 for the application assertion.

NOTE The items related by SELF may be either local instances or external instances; WHERE rules ensure that either the local XOR the external instances exist.

4.2.133.5 item_2

The item_2 specifies the related item of the relationship, in case it is in the same instance model as itself. See 4.3.80 for the application assertion.

4.2.134 Item_structure

An Item_structure is a type of both Definable_object (see 4.2.56) and Versionable_object (see 4.2.307) that is a container being able to hold Items (see 4.2.132) possibly related by Item_ relationships (see 4.2.133). An Item_structure forms a graph without any restriction neither regarding the number of entries, the connectivity nor the cyclicity. Each entity that is intended to collect Items and their relationships in a specified context should inherit from Item_structure.

The data associated with an Item_structure are the following:

- external_items;
- external_relationships;
- items;
- relationships.

4.2.134.1 external_items

The `external_items` specifies the Items belonging locally or externally to SELF. See 4.3.81 for the application assertion.

4.2.134.2 external_relationships

The `external_relationships` specifies the relationships (local and external ones) between the Items belonging to SELF. See 4.3.81 for the application assertion.

4.2.134.3 items

The `items` specifies the Items belonging locally to itself. See 4.3.82 for the application assertion.

4.2.134.4 relationships

The `relationships` specifies the relationships (local) between the Items belonging to itself. See 4.3.83 for the application assertion.

4.2.135 Length_measure_with_unit

A `Length_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is length.

4.2.136 Length_unit

A `Length_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which distances are measured.

4.2.137 Library_definition

A `Library_definition` is a type of `Definition` (see 4.2.57) that is the SUPERTYPE for all kinds of library definitions.

The data associated with a `Library_definition` are the following:

— `source`.

The `source` specifies the reference to the object in a library. See 4.3.84 for the application assertion.

4.2.138 Library_design_definition

A `Library_design_definition` is a type of both `Design_Definition` (see 4.2.63) and `Library_definition` (see 4.2.137) that is a definition specified within a library and that is concerned with the life-cycle phase of design.

4.2.139 Library_element_reference

A `Library_element_reference` provides the means to reference information about a class within a P-Lib dictionary.

The data associated with a `Library_element_reference` are the following:

- `library_identifier`;
- `property_value_pairs`.

4.2.139.1 library_identifier

The `library_identifier` specifies the identification of the component within a P-Lib Library. See 4.3.85 for the application assertion.

4.2.139.2 property_value_pairs

The `property_value_pairs` specifies the set of pairs of (property BSU, property value) defining the properties of the class. See 4.3.86 for the application assertion.

4.2.140 Library_manufacturing_definition

A `Library_manufacturing_definition` is a type of both `Design_Definition` (see 4.2.154) and `Library_definition` (see 4.2.137) that is a definition specified within a library and that is concerned with the life-cycle phase of manufacturing.

4.2.141 Library_profile_cross_section

A `Library_profile_cross_section` is a type of `Profile_cross_section` (see 4.2.206) that is the reference to a standard profile cross section standardized in an ISO 13584 library.

4.2.142 Lightship_definition

A `Lightship_definition` is a type of `Definition` (see 4.2.57) that is the definition of the weight of the ship's hull structure, including the weight of any installed machinery and outfitting, but excluding the weight of the crew, any passengers and cargoes.

The data associated with a `Lightship_definition` are the following:

- `defined_for`;
- `lightship_centre_of_gravity`;
- `lightship_items`;

— `lightship_weight`.

4.2.142.1 defined_for

The `defined_for` specifies the redefinition of the `defined_for` attribute, a `Lightship_definition` is only valid for types of Ships. See 4.3.89 for the application assertion.

4.2.142.2 lightship_centre_of_gravity

The `lightship_centre_of_gravity` specifies the centre of gravity of the lightweight in the ships global coordinate system.

4.2.142.3 lightship_items

The `lightship_items` specifies the components which make up the lightweight definition. See 4.3.88 for the application assertion.

4.2.142.4 lightship_weight

The `lightship_weight` specifies the lightweight of the ship expressed in units of mass.

4.2.143 Lightship_weight_item

A `Lightship_weight_item` is a type of `Weight_and_centre_of_gravity` (see 4.2.317) that is a component which is a part of the total lightship weight. It is be a part of the ship's hull structure, machinery or outfitting, but not cargo, crew or passengers.

The data associated with a `Lightship_weight_item` are the following:

- `aft_weight_extent`;
- `fwd_weight_extent`;
- `lightship_item_description`.

4.2.143.1 aft_weight_extent

The `aft_weight_extent` specifies the length ordinate in the `Lightship_weight_item` local co-ordinate system identifying the aft extent of the effect of the `Lightship_weight_item`.

4.2.143.2 fwd_weight_extent

The `fwd_weight_extent` specifies the length ordinate in the `Lightship_weight_item` local co-ordinate system identifying the forward extent of the effect of the `Lightship_weight_item`.

4.2.143.3 lightship_item_description

The `lightship_item_description` specifies the descriptive label of the `Lightship_weight_item`.

4.2.144 Linear_stiffness_measure_with_unit

A `Linear_stiffness_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is the ability of resistance against linear movement.

4.2.145 Linear_stiffness_unit

A `Linear_stiffness_unit` is a type of `Derived_unit` (see 4.2.61) that is the unit of the force by which a linear movement is resisted.

4.2.146 Loadline

A Loadline is the line on the ship indicating the depth to which it sinks in the water when properly loaded (see also the definition in SOLAS, clause 3).

The data associated with a Loadline are the following:

- `load_line_block_coefficient`;
- `load_line_depth`;
- `load_line_displacement`;
- `load_line_draught`;
- `load_line_length`;
- `load_line_regulation`.

4.2.146.1 load_line_block_coefficient

The `load_line_block_coefficient` is the ratio measure obtained when the `load_line_displacement` is divided by the product of `load_line_length`, `load_line_depth` and the moulded breadth of the ship.

4.2.146.2 load_line_depth

The `load_line_depth` is the least moulded depth of the ship.

4.2.146.3 load_line_displacement

The `load_line_displacement` is the moulded displacement of the ship measured at the `load_line_draught`.

4.2.146.4 load_line_draught

The load_line_draught is the draught corresponding to 0.85 times the load_line_depth of the ship

4.2.146.5 load_line_length

The load_line_length is the length of the ship measured in accordance with the applicable IMO Load Line Convention.

4.2.146.6 load_line_regulation

The load_line_regulation is a reference to the relevant IMO Load Line Convention.

4.2.147 Local_co_ordinate_system

A Local_co_ordinate_system is a type of both Definition (see 4.2.57) and axis2_placement_3d (see ISO 10303, Part 42) that is used to locate something in space. A Local_co_ordinate_system is always defined with respect to another co-ordinate system, this might be the global co-ordinate system or another Local_co_ordinate_system which has to member in the same hierarchy.

NOTE Local axes and origin are handled in the same way as for axis2_placement_3d.

NOTE A local_co_ordinate system shall form always a right handed system.

The data associated with a Local_co_ordinate_system are the following:

— parent.

The parent specifies the underlying coordinate system which serves as definition space for "this" coordinate system.

4.2.148 Local_co_ordinate_system_with_position_reference

A Local_co_ordinate_system_with_position_reference is a type of Local_co_ordinate_system (see 4.2.147) that is a special Local_co_ordinate system which directly refers to the unique Global_axis_placement as its parent. Its location is defined by references to longitudinal, vertical or transversal frames, possibly using an additional offset value (a distance). Alternatively absolute coordinates may be specified. Also, combinations of coordinates and references are valid. A Local_co_ordinate_system_with_position_reference shall not specify rotations as transformation to the global system, i.e. its axes are required to be parallel to the axes of the Global_axis_placement.

The data associated with a Local_co_ordinate_system_with_position_reference are the following:

— location;

- longitudinal_ref;
- transversal_ref;
- vertical_ref.

4.2.148.1 location

The location specifies the origin of the local co_system, defined in the parent Co_ordinate_system. derived from local or possibly global definition. See 4.3.90 for the application assertion.

4.2.148.2 longitudinal_ref

The longitudinal_ref refers to a Longitudinal_position, possibly with an offset value or an absolute coordinate value along the longitudinal axis of the global co-ordinate system. Each longitudinal_ref may be one of the following: Spacing_position (see 4.2.246), Length_measure. See 4.3.91 for the application assertion.

4.2.148.3 transversal_ref

The transversal_ref refers to a Transversal_position, possibly with an offset value or an absolute coordinate value along the transversal axis of the global co-ordinate system. Each transversal_ref may be one of the following: Spacing_position (see 4.2.246), Length_measure. See 4.3.91 for the application assertion.

4.2.148.4 vertical_ref

The vertical_ref refers to a Vertical_position, possibly with an offset value or an absolute coordinate value along the vertical axis of the global co-ordinate system. Each vertical_ref may be one of the following: Spacing_position (see 4.2.246), Length_measure. See 4.3.91 for the application assertion.

4.2.149 Logical_property_value

A Logical_property_value is a type of Property_value (see 4.2.216) that provides a value for a property of type LOGICAL.

The data associated with a Logical_property_value are the following:

- val.

The val specifies the value of type LOGICAL.

4.2.150 Longitudinal_position

A Longitudinal_position is a type of Spacing_position (see 4.2.246) that is a spacing position located on the global x axis.

4.2.151 Longitudinal_table

A Longitudinal_table is a type of Spacing_table (see 4.2.248) that is a spacing table whose positions lie on the longitudinal axis of the global co-ordinate system, which is the global x axis.

The data associated with a Longitudinal_table are the following:

— spacing_table_representations.

The spacing_table_representations specifies the longitudinal positions which make up the longitudinal table. See 4.3.92 for the application assertion.

4.2.152 Luminous_intensity_measure_with_unit

A Luminous_intensity_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a luminous_intensity as defined in ISO 31 (clause 2).

4.2.153 Luminous_intensity_unit

A Luminous_intensity_unit is a type of Named_unit (see 4.2.164) that is the unit in which the brightness of a body is measured.

4.2.154 Manufacturing_definition

A Manufacturing_definition is a type of Definition (see 4.2.57) that is the supertype for all kinds of manufacturing definitions.

The data associated with a Manufacturing_definition are the following:

— representations.

The representations specifies the representations of the manufacturing definition of a Definable_object. It is possible for a Manufacturing_definition to have multiple representations. See 4.3.93 for the application assertion.

4.2.155 Mass_measure_with_unit

A Mass_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a mass as defined in ISO 31 (clause 2).

4.2.156 Mass_unit

A `Mass_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which the amount of matter that a body contains is measured.

4.2.157 Maximum_permissible_still_water_bending_moment

A `Maximum_permissible_still_water_bending_moment` is a type of `Bending_moment` (see 4.2.14) that is the maximum valid bending moment for a ship under still water conditions as assigned by the classification society. These values will change for short voyages and sheltered water conditions. However, these can be derived from the sea-going conditions and so are not necessary for a model used for exchange purposes.

The data associated with a `Maximum_permissible_still_water_bending_moment` are the following:

- `table_of_max_values`.

The `table_of_max_values` specifies the set of maximum hogging and sagging moments at specified positions along the ship. See 4.3.94 for the application assertion.

4.2.158 Maximum_permissible_still_water_shear_force

A `Maximum_permissible_still_water_shear_force` is a type of `Shear_force` (see 4.2.239) that is the maximum valid shear force for a ship under still water conditions as assigned by the classification society. These values will change for short voyages and sheltered water conditions. However, these can be derived from the sea-going conditions and so are not necessary for a model used for exchange purposes.

The data associated with a `Maximum_permissible_still_water_shear_force` are the following:

- `table_of_max_shear_force_values`.

The `table_of_max_shear_force_values` specifies the set of values that defines the maximum permissible shear force distribution along the ship. See 4.3.95 for the application assertion.

4.2.159 Maximum_permissible_SWBM_values

The `Maximum_permissible_SWBM_values` are maximum values for hogging and sagging moments at specified positions along the ship. The values shall represent classification society requirements.

The data associated with a `Maximum_permissible_SWBM_values` are the following:

- `hogging`;
- `position`;

— sagging.

4.2.159.1 hogging

The hogging specifies the maximum hogging value.

4.2.159.2 position

The position specifies the distance from the aft-perpendicular from where the sagging and hogging moments are taken, given as a frame or station position. See 4.3.96 for the application assertion.

4.2.159.3 sagging

The sagging specifies the maximum sagging value.

4.2.160 Maximum_permmissible_SWSF_values

The Maximum_permmissible_SWSF_values are maximum values for positive and negative shear force at specified positions along the ship. The values shall represent classification society requirements.

The data associated with a Maximum_permmissible_SWSF_values are the following:

- negative_value;
- position;
- positive_value.

4.2.160.1 negative_value

The negative_value specifies the maximum negative shear force value that contributes to overall shear.

4.2.160.2 position

The position specifies the distance from the aft-perpendicular from where the shear forces are taken, given as a frame or station position. See 4.3.97 for the application assertion.

4.2.160.3 positive_value

The positive_value specifies the maximum positive shear force value that contributes to overall shear.

4.2.161 Moment_3d

A `Moment_3d` is declared to denotes a collection of moment components at three major co-ordinate system axes, vertical, transverse and longitudinal.

The data associated with a `Moment_3d` are the following:

- `longitudinal_moment`;
- `origin`;
- `transverse_moment`;
- `vertical_moment`.

4.2.161.1 longitudinal_moment

The `longitudinal_moment` specifies the moment component at the longitudinal axe.

4.2.161.2 origin

The `origin` specifies the point in the local co-ordinate system about which the moment component's lever arms were computed. See 4.3.98 for the application assertion.

4.2.161.3 transverse_moment

The `transverse_moment` specifies the moment component at the transverse axe.

4.2.161.4 vertical_moment

The `vertical_moment` specifies the moment component at the vertical axe.

4.2.162 Moment_measure_with_unit

A `Moment_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is a moment.

4.2.163 Moment_unit

A `Moment_unit` is a type of `Derived_unit` (see 4.2.61) that is the unit of the physical quantity of moment, which is the product of a force and its distance to the point of moment.

4.2.164 Named_unit

A `Named_unit` is a unit quantity associated with the word, or group of words, by which the unit is identified. A `Named_unit` may be one of following: a `Area_unit` (see 4.2.8), a `Force_unit` (see

4.2.106), a Frequency_unit (see 4.2.111), a Length_unit (see 4.2.136), a Luminous_intensity_unit (see 4.2.153), a Mass_unit (see 4.2.156), a Plane_angle_unit (see 4.2.183), a Pressure_unit (see 4.2.199), a Ratio_unit (see 4.2.219), a Time_unit (see 4.2.296), a Thermodynamic_temperature_unit (see 4.2.294), a Voltage_unit (see 4.2.312), a Volume_unit (see 4.2.314).

The data associated with a Named_unit are the following:

- dimensions.

The dimensions specify the exponents of the base properties by which the Named_unit is defined. See 4.3.99 for the application assertion.

4.2.165 Navy_Ship

A Navy_Ship is a type of ShipType (see 4.2.243) that is a ship operating under military command.

The data associated with a Navy_Ship are the following:

- has_type.

The has_type specifies the type of naval ships.

The value of the has_type shall be one of the following:

- Aircraft carrier;
- Auxiliary oiler;
- Corvette;
- Cruiser;
- Destroyer;
- Fleet auxiliary vessel;
- Frigate;
- Landing platform dock;
- Landing platform helicopter;
- Mine warfare ship;
- Patrol force vessel;

- Service craft;
- Submarine;
- User defined;

NOTE See 4.2.165.2.1 - 4.2.165.2.14 for the definition of each allowable value for `has_type`.

4.2.165.1

Aircraft carrier

the role of an aircraft carrier, apart from providing a command, control and communications facility, is to deploy aircraft (either `stovl`, `stobar` or `ctol`) and/or helicopters in sustained anti-submarine warfare operations and fighter protection, with full reconnaissance and strike capability

4.2.165.2

Auxiliary oiler

a ship specially designed to replenish ships at sea with liquids during world-wide operations, with VERTREP services for the transfer of solids also being provided. The ship has a stable platform suitable for helicopters, including stowage and maintenance facilities plus emergency landing of other helicopters

4.2.165.3

Corvette

a ship specifically constructed for the purpose of performing the role of a small escort vessel to a task group. Corvettes are fitted primarily to fulfill an anti-submarine warfare role

4.2.165.4

Cruiser

a ship typically used to screen carrier task forces. Cruisers (many with guided missiles or carrying a helicopter) provide anti-air warfare and anti-submarine capabilities. Cruisers also provide protection against anti-ship cruise missile threats at extended ranges, particularly in the presence of enemy electronic countermeasures

4.2.165.5

Destroyer

a major surface combatant which is typically used to conduct operations with strike, anti-submarine warfare and amphibious forces, and to perform screening and convoy duties. A destroyer may also be equipped with helicopters, providing an enhanced capability

4.2.165.6

Fleet auxiliary vessel

the primary task of a fleet auxiliary vessel (of which there are many types) is to supply warships at sea with fuel, food, stores and ammunition. An auxiliary also provides aviation platforms, amphibious support for the navy and marines and sea transport for army units

4.2.165.7

Frigate

a ship that can be generally described as a general purpose, ocean escort vessel. Operational requirements necessitate that frigates perform the duties of area defense ships, capable of defending a task group against modern day air threats. Frigates may also provide command facilities and accommodation. Secondary capabilities include anti-surface warfare, naval gunfire support and anti-submarine warfare

4.2.165.8

Landing platform dock

a ship specifically designed to transport a large embarked military force (plus support equipment) across open oceans; support a flexible landing on hostile shores using on-board helicopters and landing craft, and co-ordinate the naval, air and land aspects of amphibious operations through Command, Control and Communications facilities

4.2.165.9

Landing platform helicopter

an amphibious helicopter carrier designed to enable the rapid tactical deployment of airborne troops and equipment to spearhead amphibious operations ashore. It can also stand off a coast at strategic range in a deterrent operational role. Peacetime roles include troop and equipment transport, and humanitarian tasks such as disaster relief

4.2.165.10

Mine warfare ship

specially constructed for the hunting, sweeping (by mechanical, acoustic or magnetic means) and clearance of mines in both inshore/coastal and deep, exposed waters. Designed to have minimal magnetic signature, MCMV hulls are generally constructed of low-magnetic steel, GRP or laminated wood

4.2.165.11

Patrol force vessel

the size of patrol force vessels can vary according to the roles for which they have been designed. Larger offshore patrol vessels (OPVs) are used for firefighting, rescue or supply tasks, others are used as protection vessels, sometimes carrying a marine detachment and semi-rigid craft to act as a rapid response squadron. Some OPVs have ice-strengthened hulls and a helicopter landing deck for operation as survey vessels in the Arctic or Antarctic regions. The size, maneuverability and other operational characteristics of smaller coastal patrol craft means that they are ideally suited for patrol, search and rescue duties in coastal areas, or ports, harbors and other restricted waters

4.2.165.12

Service craft

the primarily small vessels that have been specially designed to provide services to a fleet in harbors and ports. These include tugs, tenders, barges, patrol craft, recovery vessels, floating docks, etc. Other service craft providing support are larger ocean-going vessels, such as transports, survey and research ships, repair vessels, cargo ships, hospital ships, etc

4.2.165.13**Submarine**

a high-capability, ocean-going vessel designed typically to perform both anti-submarine and anti-surface warship tasks. To perform these functions, generally a submarine's weapon system is capable of stowing, firing and controlling heavyweight torpedoes, underwater-to-surface guided weapons or submarine-laid mines. Submarines are able to operate submerged in shallow waters as well as the open ocean, whilst also being capable of operation environmentally in waters from the tropics to the arctic

4.2.165.14**User defined**

a user defined naval ship type

4.2.166 Non_circular_hollow_profile_cross_section

A `Non_circular_hollow_profile_cross_section` is a type of `Parametric_profile_cross_section` (see 4.2.178) identified by having a hollow core and closed perimeter.

NOTE ISO 657-14 specifies dimensions for some standard square and rectangular hollow sections.

4.2.167 Number_property_value

A `Number_property_value` is a type of `Property_value` (see 4.2.216) that provides a value for a property of type `NUMBER`.

The data associated with a `Number_property_value` are the following:

— `val`.

The `val` specifies the value of type `NUMBER`.

4.2.168 Outward_round_corner_design_definition

An `Outward_round_corner_design_definition` is a type of `Round_corner_design_definition` (see 4.2.229) that specifies the radius and location the centre of the circle arc representing the outwards round corner. The centre of the circle arc is placed inside the work item with the local origin located at the corner that shall be removed. The transition between the circle arc (surface) and the adjacent edges may be smooth.

The data associated with an `Outward_round_corner_design_definition` are the following:

— `x_offset`;

— `y_offset`.

4.2.168.1 x_offset

The x_offset specifies the distance of the centre of the circle arc in positive x-direction.

4.2.168.2 y_offset

The y_offset specifies the distance of the centre of the circle arc in positive y-direction.

4.2.169 Owner_designation

An Owner_designation is a type of General_characteristics_definition (see 4.2.113) that specifies the organizations that order, own and manage the ship.

The data associated with an Owner_designation are the following:

- local_units;
- managing_company;
- ordering_company;
- owning_company.

4.2.169.1 local_units

The local_units specifies that an Owner_designation shall not define local units.

4.2.169.2 managing_company

The managing_company specifies the organization that is responsible for managing and operating the ship. See 4.3.100 for the application assertion.

4.2.169.3 ordering_company

The ordering_company specifies the organization that placed an order with a shipyard for the ship. See 4.3.100 for the application assertion.

4.2.169.4 owning_company

The owning_company specifies the organization that legally owns the ship. See 4.3.100 for the application assertion.

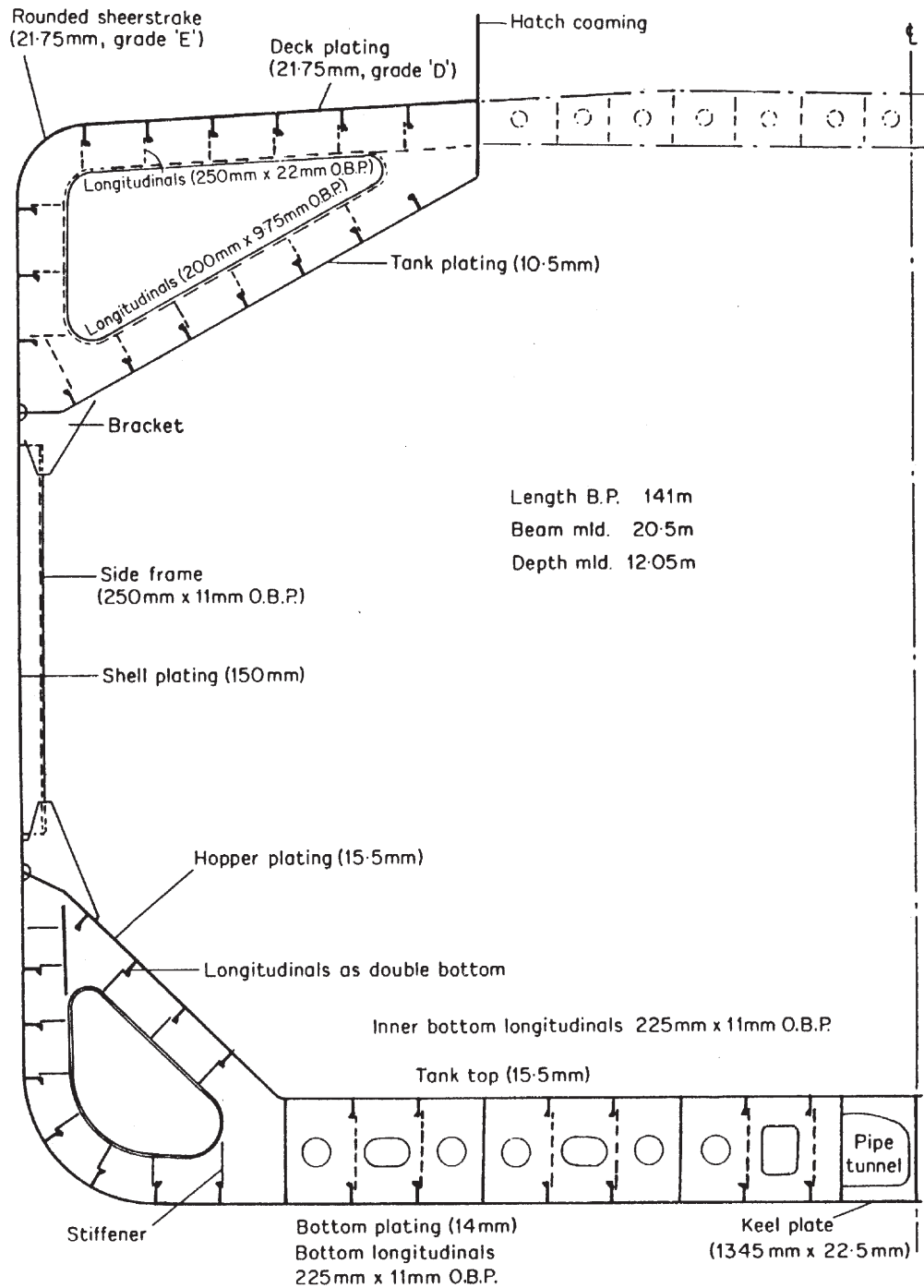


Figure 10 – A hull cross section of a bulk carrier

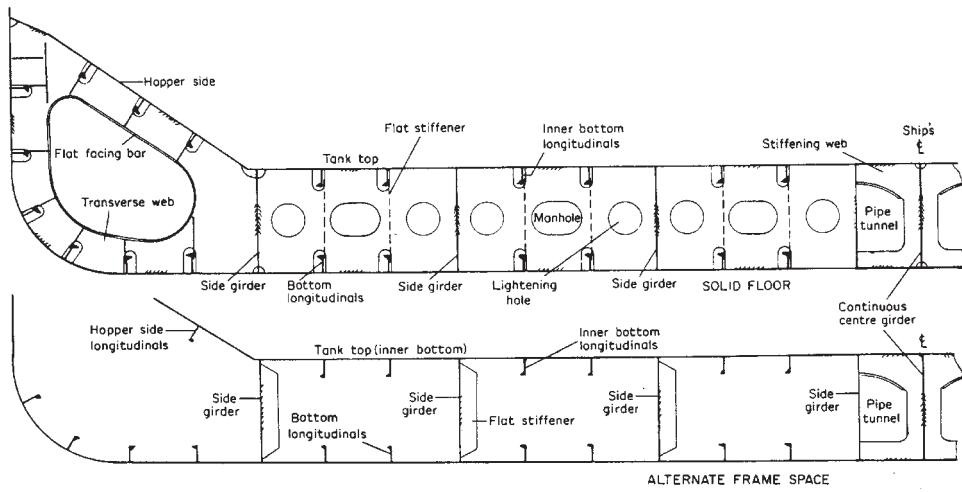


Figure 11 – A doublebottom of a bulk carrier

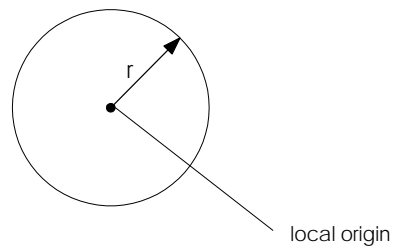


Figure 12 – Circular_cutout

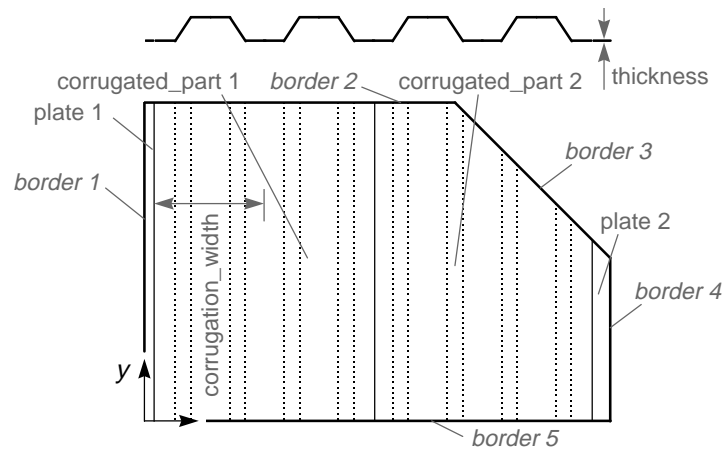


Figure 13 – Corrugated system parameters

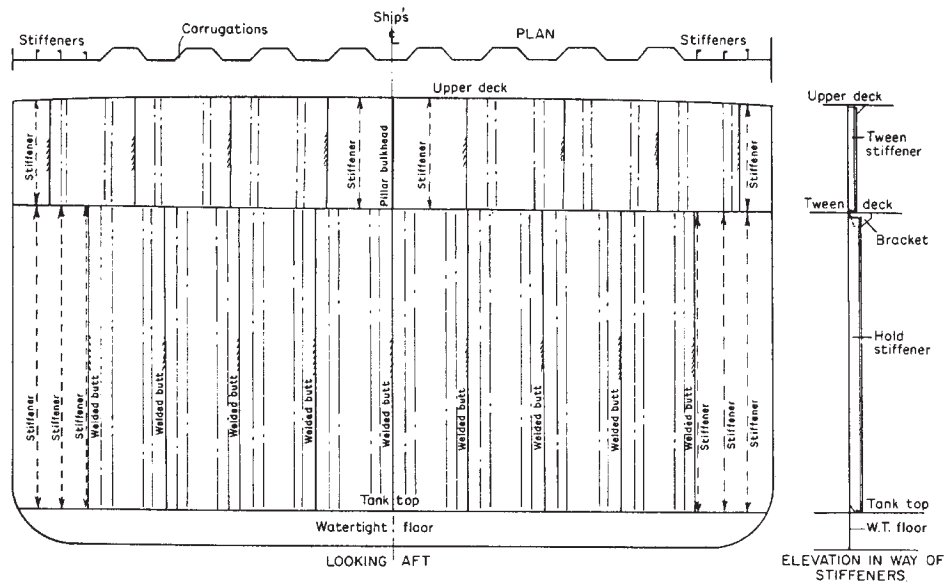


Figure 14 – A corrugated bulkhead

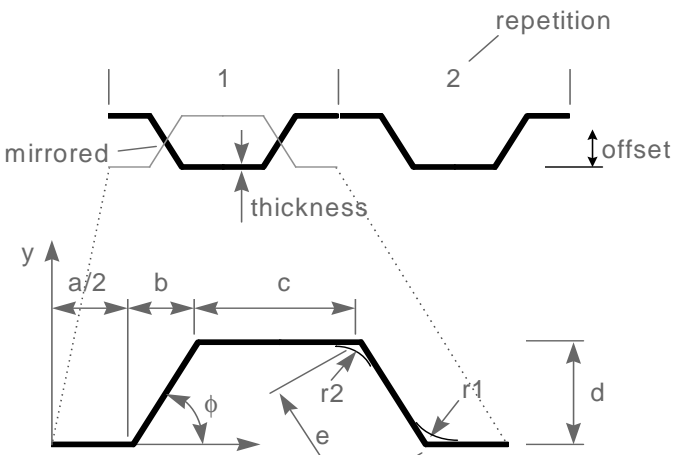


Figure 15 – Corrugated part parameters

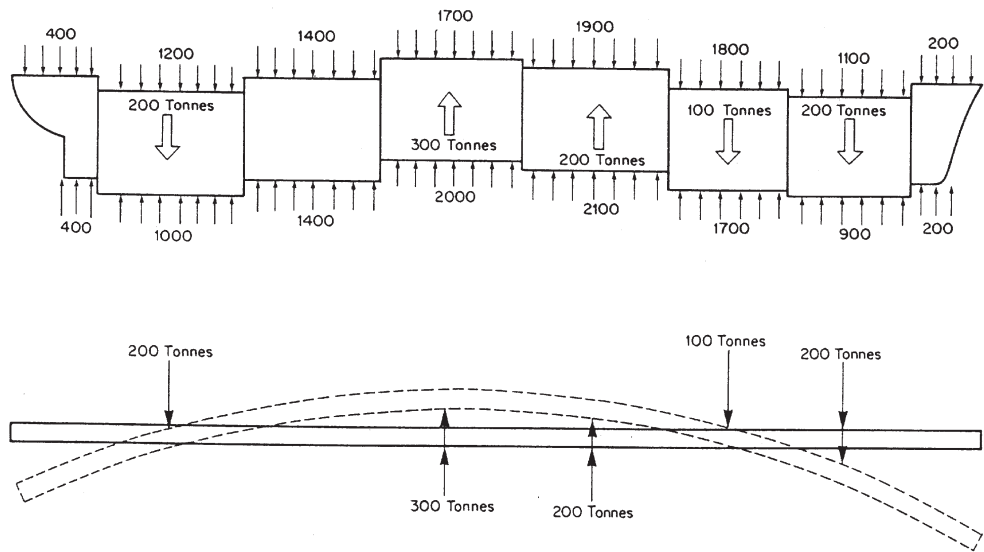


Figure 16 – Design still water bending moment

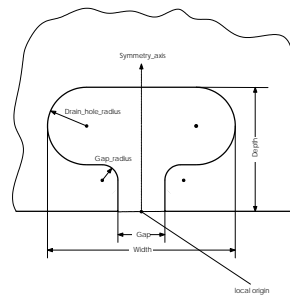


Figure 17 – Drain_hole_cutout

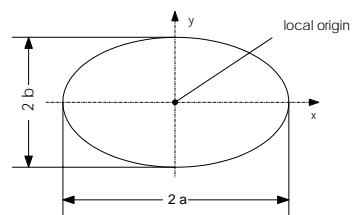


Figure 18 – Elliptical_cutout

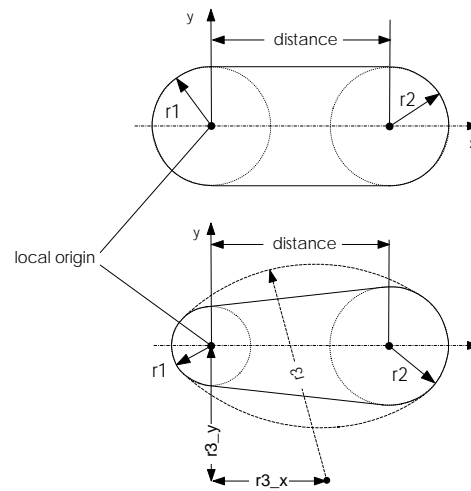


Figure 19 – Elongated_oval_cutout

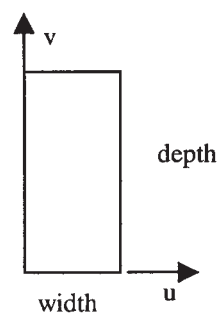


Figure 20 – A Flat_bar_cross_section

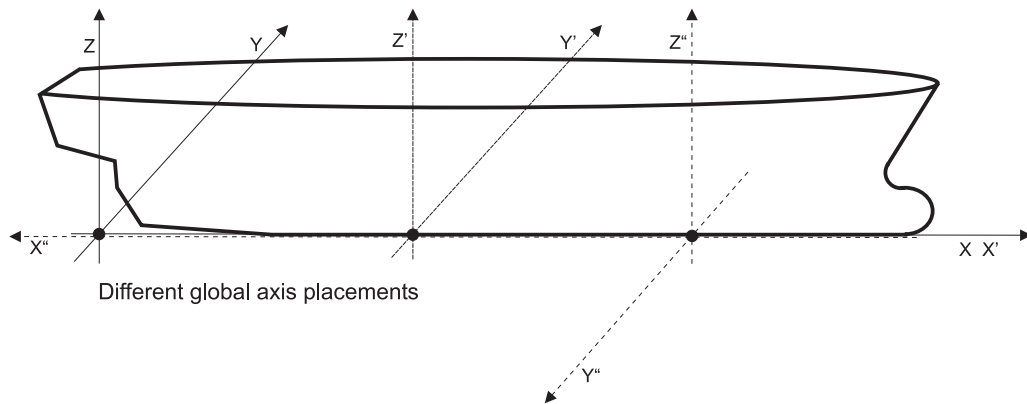


Figure 21 – Global axis placements

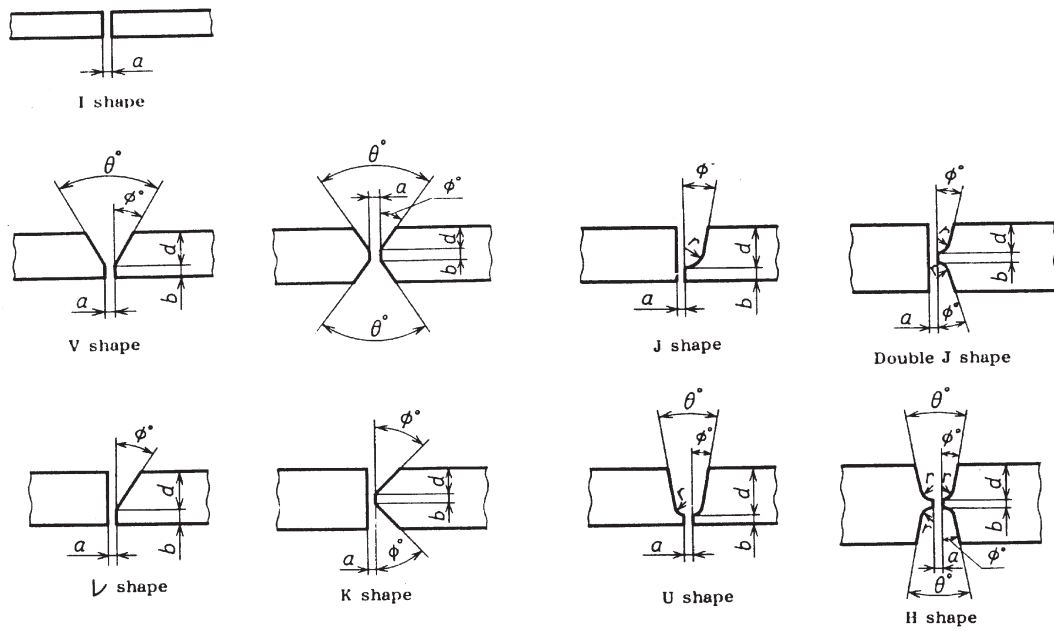


Figure 22 – An example of groove weld

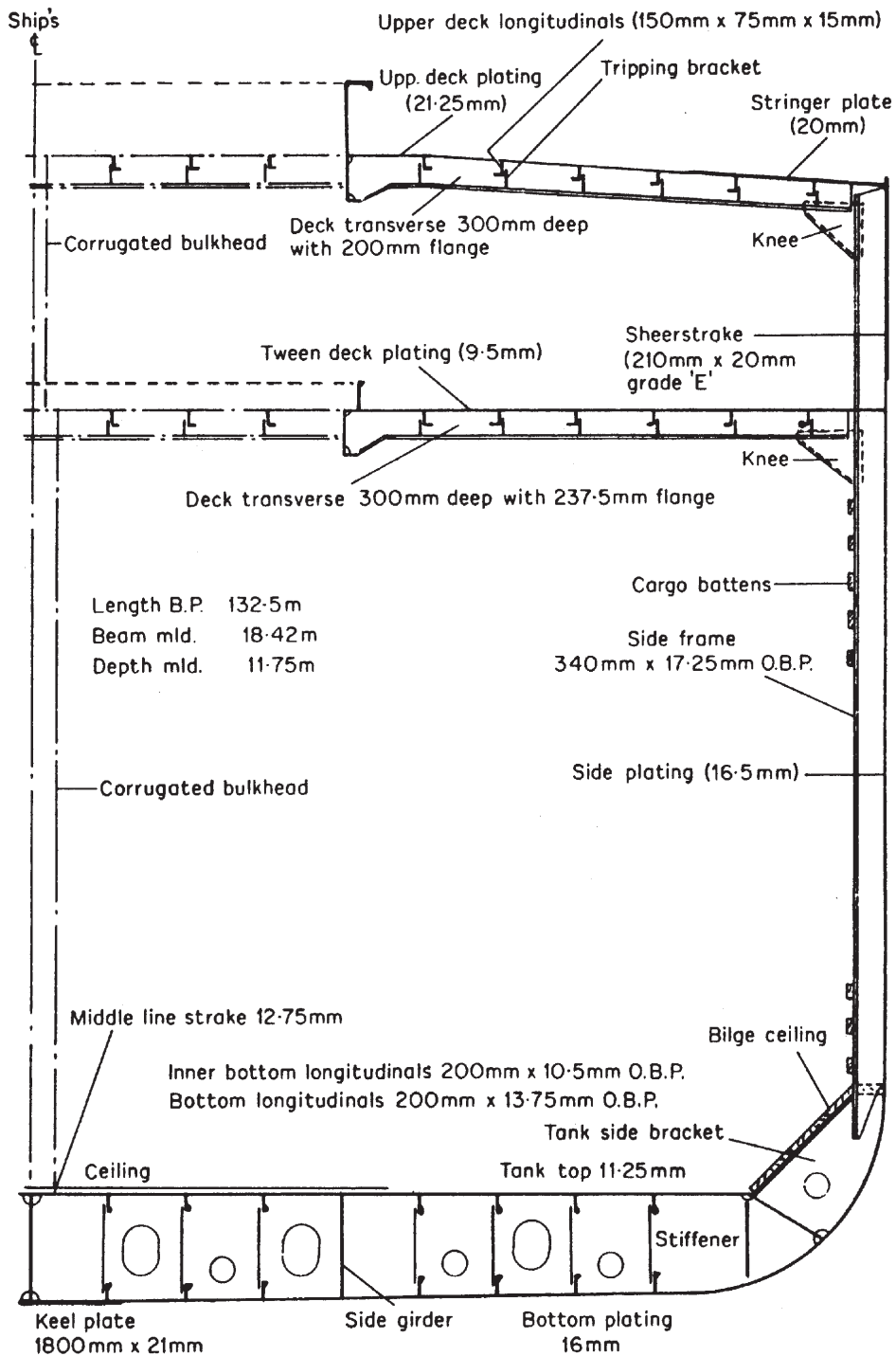


Figure 23 – A hull cross section of a cargo ship

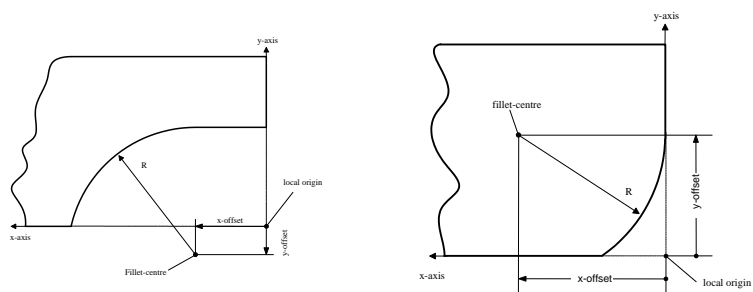


Figure 24 – Inward_round_corner_cutout and Outward_round_corner_cutout

4.2.170 Panel_system

A Panel_system is a type of Structural_system (see 4.2.279) that is a special section that consists of one to many Plates (see 4.2.184) reinforced by Profiles running in the same main direction; the Plates lie in the same (smooth mathematical) moulded surface (plane or non-plane).

The data associated with a Panel_system are the following:

— class.

The class specifies the class of a Panel_system needs to be PANEL_SYSTEM.

4.2.171 Panel_system_boundary_relationship

A Panel_system_boundary_relationship is a type of Panel_system_relationship (see 4.2.176) that defines a boundary of a Panel_system (see 4.2.170) by referring to a "structural item". "Structural_item" can be one of the following:

Panel_system the boundary is created by the intersection of the (implicitly or explicitly) given moulded surfaces of the two Panel_systems;

Profile the trace line of the Profile must either be parallel or orthogonal to the (implicitly or explicitly) given moulded surface of SELF. In the first case the boundary is created by the intersection of the Profile top line with the moulded surface of SELF, in the second case it is created by the the web of the cross section of the Profile;

Seam the Seam is the boundary.

The data associated with a Panel_system_boundary_relationship are the following:

— displacement.

The displacement specifies the offset orthogonally off the referenced boudary.

4.2.172 Panel_system_curve_boundary

A Panel_system_curve_boundary is the "just geometry" type of boundary for a Panel_system (see 4.2.170).

The data associated with a Panel_system_curve_boundary are the following:

— curve;

— displacement.

4.2.172.1 curve

The curve defines the boundary geometry. See 4.3.101 for the application assertion.

4.2.172.2 displacement

The displacement specifies the displacement specifying the offset of the boundary orthogonal off the curve.

4.2.173 Panel_system_design_definition

A `Panel_system_design_definition` is a type of `Structural_system_design_definition` (see 4.2.281) that describes the definition of a `Panel_system` (see 4.2.170) for the life-cycle stage preliminary design.

The data associated with a `Panel_system_design_definition` are the following:

- `border`;
- `defined_for`;
- `material_offset`;
- `moulded_surface`;
- `stiffener_spacing`;
- `thickness`.

4.2.173.1 border

The border specifies the border of a `Panel_system` is made of a number of (at least 3) boundaries. The order of the boundaries shall be significant but there is no rule that requires $endpointofboundary[n] = startpointofboundary[n + 1]$. Furthermore, two boundaries may intersect or it might be necessary to (straight) elongate one or both boundaries in order to make them intersecting; boundary can be one of

- a) a curve parallel to a given curve with a given displacement;
- b) a line resulting from the intersection of the `Panel_system`'s x-y-plane with a given plane;
- c) a line resulting from the intersection of the `Panel_system`'s x-y-plane with the x-y-plane of a given `Panel_system`;
- d) a curve parallel to a boundary of a given `Panel_system` with a given displacement;

- e) a curve parallel to the top line of a given Profile with a given displacement;
- f) a line parallel to the mould line of the web of the cross section of a given Profile with a given displacement;
- g) a seam.

4.2.173.2 defined_for

The `defined_for` specifies that the `Panel_design_definitions` shall only be declared for `Panel_systems`. See 4.3.103 for the application assertion.

4.2.173.3 material_offset

The `material_offset` specifies the offset information for the location of the panel material with respect to the moulded surface (implicitly or explicitly given) that is valid for the panel in fractions of the material thickness; to be interpreted as follows: `material_offset` 100% of the material located on the positive side of the moulded surface.

EXAMPLE 1 *material_offset* < 0: material located completely below the negative side of the moulded surface;

material_offset = 0: material located completely on the negative side of the moulded surface;

0 < *material_offset* < 1: material located in the moulded surface;

material_offset = 1: material located completely on the positive side of the moulded surface;

material_offset > 1: material located completely above the positive side of the moulded surface.

NOTE The `material_offset` specified here serves as a reference value for the `material_offset` of the Plates defined for the `Panel_system` referred to in a later stage of the design. For some reason it may happen that the `material_offset` of the Plates differs from the one specified here.

4.2.173.4 moulded_surface

The `moulded_surface` specifies the underlying moulded surface of the `Panel_system` (if not specified the local x-y-plane is implicitly assumed as underlying moulded surface). The `moulded_surface` need not to be specified for a particular `Panel_system_design_definition`.

4.2.173.5 stiffener_spacing

The `stiffener_spacing` specifies the proposed spacing of the stiffeners belonging to the panel (OPTIONAL because not all `Panel_systems` have a meaningful `stiffener_spacing`).

NOTE The `stiffener_spacing` specified here serves as a reference value for the spacing of the Profiles defined on the `Panel_system` referred to in a later stage of the design. For some reason it may happen that the spacing of the Profiles differs from the one specified here.

The `stiffener_spacing` need not to be specified for a particular `Panel_system_design_definition`.

4.2.173.6 thickness

The `thickness` specifies the proposed material thickness of the `Panel_system`.

NOTE The `thickness` specified here serves as a reference value for the thickness of the Plates defined for the `Panel_system` referred to in a later stage of the design. For some reason it may happen that the thickness of the Plates differs from the one specified here.

4.2.174 Panel_system_functional_definition

A `Panel_system_functional_definition` is a type of `Structural_system_functional_definition` (see 4.2.282) that defines the function of a `Panel_system` (see 4.2.170).

The data associated with a `Panel_system_functional_definition` are the following:

— `defined_for`.

The `defined_for` specifies the a `Panel_system_functional_definition` shall only be valid for `Panel_systems`. See 4.3.103 for the application assertion.

4.2.175 Panel_system_plane_boundary

A `Panel_system_plane_boundary` specifies a boundary by intersecting with the (implicitly or explicitly) given moulded surface of a `Panel_system` (see 4.2.170).

The data associated with a `Panel_system_plane_boundary` are the following:

— `plane`.

The `plane` specifies the defines the boundary geometry by intersecting with the (implicitly or explicitly) given moulded surface of a `Panel_system`. See 4.3.104 for the application assertion.

4.2.176 Panel_system_relationship

A `Panel_system_relationship` is a type of `Structural_system_relationship` (see 4.2.284) that describes the association of a `Panel_system` (see 4.2.170) with another "Structural item". A "Structural item" can be one of

- a) `Structural_system`;
- b) `Structural_part`;
- c) `Structural_feature`.

The data associated with a `Panel_system_relationship` are the following:

— `item_1`.

The `item_1` specifies the `Panel_system_relationship` shall relate a `Panel_system` with something else. See 4.3.105 for the application assertion.

4.2.177 Parametric_definition

A `Parametric_definition` is a type of `Definition` (see 4.2.57) that is the supertype for all kinds of parametric definitions

4.2.178 Parametric_profile_cross_section

A `Parametric_profile_cross_section` is a type of `Profile_cross_section` (see 4.2.206). A `Parametric_profile_cross_section` is either a `Flanged_profile_cross_section` (see 4.2.100), or a `Channel_profile_cross_section` (see 4.2.32), a `Bar_profile_cross_section` (see 4.2.15), an `Angle_profile_cross_section` (see 4.2.2), a `Non_circular_hollow_profile_cross_section` (see 4.2.166), a `Circular_hollow_profile_cross_section` (see 4.2.35), or a `Proprietary_profile_cross_section` (see 4.2.217). A `Parametric_profile_cross_section` is a standard shape that can be defined by specification of values for a set of pre-defined geometric parameters.

4.2.179 Part

A `Part` is a type of `Item` (see 4.2.132) that is the atomic element a `System` (see 4.2.289) and a `Assembly` (see 4.2.9) consist of. It is an `Item` (see 4.2.132) (to be) created by a physical activity and (to be) made of a material.

4.2.180 Part_edge_cutout_design_definition

A `Part_edge_cutout_design_definition` is a type of `Feature_design_definition` (see 4.2.96) that specifies the removal of material along an edge due to the penetration of a part. The shape of the opening is dependent on the part and has, therefore, not necessarily a symmetry axis. All measures of the `Part_cutout` are relative to the part, that is, they depend on the part ID-points (specific pre-defined locations on the cross section of the part; see figure). The `x_axis` coincides with the work item edge whereas the direction of the `y_axis` is perpendicular to it. The cutout depth and inner shape depend on the part ID locations B and C. The width of the opening may depend on the part ID locations A, B, C and/or D. The radius (`R_left`, `R_right`), `x_A` and `x_D` are positive always, whereas `x_B`, `y_B`, `x_D` and `y_D` may be positive or negative.

The data associated with a `Part_edge_cutout_design_definition` are the following:

— `R_left`;

— `R_right`;

- `defined_for`;
- `x_A`;
- `x_B`;
- `x_C`;
- `x_D`;
- `y_B`;
- `y_C`.

4.2.180.1 `R_left`

The `R_left` specifies the radius of the `top_left` circle cutout with reference point B.

4.2.180.2 `R_right`

The `R_right` specifies the radius of the `top_right` circle cutout with reference point C.

4.2.180.3 `defined_for`

The `defined_for` specifies the `Edge_cutouts` that a `Part_edge_cutout_design_definition` shall be valid for. See 4.3.106 for the application assertion.

4.2.180.4 `x_A`

The `x_A` specifies the distance from reference point D along the positive direction of the x-axis.

4.2.180.5 `x_B`

The `x_B` specifies the distance and direction along the x-axis of the `top_left` circle centre from the reference point B.

4.2.180.6 `x_C`

The `x_C` specifies the distance and direction along the x-axis of the `top_right` circle centre from the reference point C.

4.2.180.7 `x_D`

The `x_D` specifies the distance from reference point A along the negative direction of the x-axis.

4.2.180.8 y_B

The y_B specifies the distance and direction along the y-axis of the top_left circle centre from the reference point B.

4.2.180.9 y_C

The y_C specifies the distance and direction along the y-axis of the top_right circle centre from the reference point C.

4.2.181 Part_relationship

A Part_relationship is a type of Item_relationship (see 4.2.133) that defines the association of a Part (see 4.2.179) with another Item.

The data associated with a Part_relationship are the following:

— item_1.

The item_1 specifies the relating Part of the relationship. See 4.3.107 for the application assertion.

4.2.182 Plane_angle_measure_with_unit

A Plane_angle_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a plane_angle as defined in ISO 31 (clause 2).

4.2.183 Plane_angle_unit

A Plane_angle_unit is a type of Named_unit (see 4.2.164) that is the unit in which angles in planes are measured.

4.2.184 Plate

A Plate is a type of Structural_part (see 4.2.269) that is made from flat shaped stock material.

4.2.185 Plate_approval

A Plate_approval is a type of Structural_class_approval (see 4.2.260) that confirms the approval of a Plate_design_definition (see 4.2.187).

The data associated with a Plate_approval are the following:

— subject

The subject specifies the reference to the approved `Plate_design_definition`. See 4.3.108 for the application assertion.

4.2.186 `Plate_boundary_relationship`

A `Plate_boundary_relationship` is a type of `Plate_relationship` (see 4.2.191) that defines a boundary of a `Plate` (see 4.2.184) by referring to a `Seam` (see 4.2.232).

The data associated with a `Plate_boundary_relationship` are the following:

- `item_2`.

The `item_2` specifies the binding Item shall be of type `Seam`. See 4.3.109 for the application assertion.

4.2.187 `Plate_design_definition`

The `Plate_design_definition` is a type of `Structural_part_design_definition` (see 4.2.271) that describes the definition of a `Plate` (see 4.2.184) from the view of detailed design.

The data associated with a `Plate_design_definition` are the following:

- `border`;
- `defined_for`;
- `material_offset`;
- `moulded_surface`;
- `thickness`.

4.2.187.1 `border`

The `border` specifies the border of a `Plate` is made of a number of (at least 3) boundaries. The order of the boundaries shall be significant but there is NO rule that requires end point of `boundary[n]` = start point of `boundary[n+1]` furthermore two boundaries may intersect or it might be necessary to (straight) elongate one or both boundaries in order to make them intersecting; boundary can be one of a curve, a `Seam` and a boundary of a `Panel_system`. Each border may be one of the following: `Panel_system_boundary`, `Plate_boundary_relationship` (see 4.2.186), `Bounded_curve` (see ISO 10303, Part 42). See 4.3.110, 4.3.112 and 4.3.114 for the application assertions.

4.2.187.2 defined_for

The `defined_for` specifies that the `Plate_design_definition` shall only be valid for Plates. See 4.3.113 for the application assertion.

4.2.187.3 material_offset

The `material_offset` specifies the offset information for the location of the plate material with respect to the moulded surface (implicitly or explicitly given) that is valid for the plate in fractions of the plate thickness; to be interpreted as follows: *material_offset* * 100% of the material located on the positive side of the moulded surface,

EXAMPLE 1 *material_offset* < 0: material located completely below the negative side of the moulded surface;

material_offset = 0: material located completely on the negative side of the moulded surface;

0 < *material_offset* < 1: material located in the moulded surface;

material_offset = 1: material located completely on the positive side of the moulded surface;

material_offset > 1: material located completely above the positive side of the moulded surface.

NOTE NOTE If the Plate belongs to a `Structural_system` that got a `Design_definition` during the preliminary design stage (like `Panel_system`) `material_offset` should match with the one specified there. For some reason however it might be necessary that a Plate gets a `material_offset` other than specified during preliminary design.

4.2.187.4 moulded_surface

The `moulded_surface` is optional and, if present, specifies the underlying moulded surface of the Plate. If not specified and `SELF` participates in a `Panel_system` the underlying moulded surface of the `Panel_system` is taken into account, otherwise the local x-y-plane is implicitly assumed as underlying moulded surface. Each `moulded_surface` may be one of the following: `Surface` (see ISO 10303, Part 42), `External_instance_reference` (see 4.2.93). See 4.3.111 and 4.3.115 for the application assertion.

4.2.187.5 thickness

The `thickness` specifies the material thickness of the plate.

NOTE If the Plate belongs to a `Structural_system` that got a `Design_definition` during the preliminary design stage (like `Panel_system`) `thickness` should match with the one specified there. For some reason however it might be necessary that a Plate gets a `thickness` other than specified during preliminary design.

4.2.188 Plate_functional_definition

A `Plate_functional_definition` is a type of `Structural_part_functional_definition` (see 4.2.272) that defines the functional role of a Plate; the role may be a pre-defined one or may be user-defined.

The data associated with a `Plate_functional_definition` are the following:

- `defined_for`.

The `defined_for` specifies that a `Plate_functional_definition` is only valid for one or multiple Plates.

4.2.189 Plate_manufacturing_definition

A `Plate_manufacturing_definition` is a type of `Structural_part_manufacturing_definition` (see 4.2.275) that is a production lifecycle stage view of a plate part that represents characteristics and attributes directly related to the manufacturing of the part. An earlier detailed design part stage view may or may not exist for the part. At this stage, structural part features are identified that affect aspects of the part that have to do with processes that will be used to cut and form the part. Some examples of the kinds of features identified during this stage are edge grinds, layout marks, and part stock. A 2D or 3D representation may be defined for the manufacturing definition of the part, enabling it to be nested and cut from flat or corrugated raw material stock. This geometry is represented by the inherited `.representation` attribute.

The data associated with a `Plate_manufacturing_definition` are the following:

- `outer_contour`.

The `outer_contour` specifies the closed curve in the representation that specifies outer boundary of the part geometry that will be burned during the manufacturing process. This is the resulting curve after the resolution of manufacturing features such added material feature and weld-shrinkage allowance. See 4.3.117 for the application assertion.

4.2.190 Plate_rejection

The `Plate_rejection` is a type of `Structural_class_rejection` (see 4.2.263) which confirms that a `Plate_design_definition` (see 4.2.187) is not approved. The reasons for rejection and existing requirements for the plate are specified.

NOTE `Plate_rejection` may specify additional information clarifying the rule requirements for the Plate. A yard has several options for action; the thickness of the plate may be increased, the material quality may be improved, another stiffening may be applied et.. Although it is not mandatory, a class society may suggest improvements in these areas. This may then be done using `Structural_class_rejection.proposed_alternative`.

The data associated with a `Plate_rejection` are the following:

- `subject`.

The subject specifies the rejected `Plate_design_definition`. See 4.3.118 for the application assertion.

4.2.191 Plate_relationship

A `Plate_relationship` is a type of `Structural_part_relationship` (see 4.2.277) that describes the relationship between a `Plate` (see 4.2.184) and another 'structural item'. 'Structural item' can be one of

- a) `Structural_system`,
- b) `Structural_part`,
- c) `Structural_feature`.

The data associated with a `Plate_relationship` are the following:

— `item_1`

The `item_1` specifies the `Plate_relationship` shall relate a `Plate` with something else. See 4.3.119 for the application assertion.

4.2.192 Plate_renewal_definition

A `Plate_renewal_definition` is a type of `Definition` (see 4.2.57) that specifies the rule and the renewal thicknesses of a `Plate`. Rule values are values that are required by e.g. classification societies for approval. These properties are specified during the newbuilding of the ship and are used during its operation, for example during classification society surveys.

The data associated with a `Plate_renewal_definition` are the following:

- `defined_for`;
- `renewal_thickness`;
- `rule_thickness`.

4.2.192.1 defined_for

The `defined_for` specifies the `Plates` that this `Definition` is valid for. See 4.3.120 for the application assertion.

4.2.192.2 renewal_thickness

The `renewal_thickness` specifies the minimum thickness of a `Plate`, below which it must be replaced or repaired.

4.2.192.3 rule_thickness

The `rule_thickness` specifies the minimum allowable thickness of a `Plate` during the life-time of the `Plate`. `Rule_thickness` is specified in the Classification Rules and Regulations.

4.2.193 Plate_strake

A `Plate_strake` is a type of `Structural_system` (see 4.2.279). It has the restriction to only consist of `Plates` and has only a tightness function but not a strength function.

The data associated with a `Plate_strake` are the following:

— `class`.

The `class` specifies the class of a `Plate_strake` needs to be `PLATE_STRAKE`.

4.2.194 Plate_strake_functional_definition

A `Plate_strake_functional_definition` is a type of `Structural_system_functional_definition` (see 4.2.282) that defines the function of a `Plate_strake` (see 4.2.193).

The data associated with a `Plate_strake_functional_definition` are the following:

— `defined_for`.

The `defined_for` specifies the a `Plate_strake_functional_definition` shall only be valid for `Plate_strakes`. See 4.3.121 for the application assertion.

4.2.195 Position_feature

A `Position_feature` is a type of `Structural_feature` (see 4.2.266). `Position_features` are used during the ship structure design process. They are intended to provide auxiliary geometry that is used to position structural elements with respect to each other or to define the geometry of not yet existing structural elements during the preliminary design phase.

4.2.196 Position_feature_design_definition

A `Position_feature_design_definition` is a type of `Feature_design_definition` (see 4.2.96) that describes the definition of a `Position_feature` from the view point of design.

The data associated with a `Position_feature_design_definition` are the following:

— `defined_for`;

— `representations`.

4.2.196.1 defined_for

The `defined_for` specifies the `Position_feature_design_definition` shall only be valid for `Position_features`. See 4.3.122 for the application assertion.

4.2.196.2 representations

The `representations` specifies the `Position_feature_design_definition` shall be represented by `Shape_representations`. See 4.3.123 for the application assertion.

4.2.197 Position_feature_relationship

A `Position_feature_relationship` is a type of `Structural_feature_relationship` (see 4.2.267) that defines the association of a `Position_feature` (see 4.2.195) with another `Item` (see 4.2.132).

The data associated with a `Position_feature_relationship` are the following:

— `item_1`.

The `item_1` specifies the relating item of the relationship shall be a `Position_feature`. See 4.3.124 for the application assertion.

4.2.198 Pressure_measure_with_unit

A `Pressure_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is a pressure.

4.2.199 Pressure_unit

A `Pressure_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which the physical quantity of force per area is measured.

4.2.200 Principal_characteristics

The `Principal_characteristics` are a type of `General_characteristics_definition` (see 4.2.113) that specify the main shape parameters of the hull moulded form. `Principal_characteristics` also includes data that is required in subsequent iterations of the hull development process when one is considering hydrostatics.

The data associated with a `Principal_characteristics` are the following:

- `block_coefficient`;
- `design_deadweight`;
- `design_draught`;

- `length_between_perpendiculars`;
- `max_draught_at_AP`;
- `max_draught_at_FP`;
- `min_draught_at_AP`;
- `min_draught_at_FP`;
- `moulded_breadth`;
- `moulded_depth`.

4.2.200.1 `block_coefficient`

The `block_coefficient` specifies the ratio of the moulded displacement volume to the volume of a block that has its length equal to the `length_between_perpendiculars`, its breadth equal to the maximum immersed `moulded_breadth` and its depth equal to the `design_draught`. The `block_coefficient` should be defined only for mono hull ships. The `block_coefficient` need not to be specified for a particular `Principal_characteristics`.

4.2.200.2 `design_deadweight`

The `design_deadweight` specifies the weight of the ship representing the weight of cargo, bunker fuel, water, passengers, crew and consumables which a ship can carry when loaded to her summer load line.

4.2.200.3 `design_draught`

The `design_draught` specifies the optimal draught to which the ship has been designed to operate.

4.2.200.4 `length_between_perpendiculars`

The `length_between_perpendiculars` specifies the length measured from the after perpendicular to the forward perpendicular of the ship.

4.2.200.5 `max_draught_at_AP`

The `max_draught_at_AP` specifies the maximum possible draught at the Aft Perpendicular during the operation of the ship. The `max_draught_at_AP` is used for at hull cross section approval for the ice class notation.

4.2.200.6 max_draught_at_FP

The max_draught_at_FP specifies the maximum possible draught at the Forward Perpendicular during the operation of the ship. The max_draught_at_FP is used at hull cross section approval for the ice class notation.

4.2.200.7 min_draught_at_AP

The min_draught_at_AP specifies the minimum possible draught at the Aft Perpendicular during the operation of the ship. The min_draught_at_AP is used at hull cross section approval for the ice class notation.

4.2.200.8 min_draught_at_FP

The min_draught_at_FP specifies the minimum possible draught at the Forward Perpendicular during the operation of the ship. The min_draught_at_FP is used at hull cross section approval for the ice class notation.

4.2.200.9 moulded_breadth

The moulded_breadth specifies the maximum breadth of the ship amidships and at the design_draught.

4.2.200.10 moulded_depth

The moulded_depth specifies the vertical distance above the baseline to the uppermost deck where the deck joins the side of the ship measured amidships.

4.2.201 Product_of_inertia_measure_with_unit

A Product_of_inertia_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a product of inertia.

4.2.202 Product_of_inertia_unit

A Product_of_inertia_unit is the unit of the physical quantity of Product_of_inertia and is measured as a length to the fifth.

4.2.203 Profile

A Profile is a type of Structural_part (see 4.2.269) that is made from profile shaped stock material.

4.2.204 Profile_approval

A Profile_approval is a type of Structural_class_approval (see 4.2.260) that confirms the approval of a Profile_design_definition (see 4.2.208).

The data associated with a Profile_approval are the following:

— subject.

The subject specifies the reference to the approved Profile_design_definition. See 4.3.125 for the application assertion.

4.2.205 Profile_boundary_relationship

A Profile_boundary_relationship is a type of Profile_relationship (see 4.2.212) that defines a boundary for the trace line of a Profile by referring to a "structural item". "Structural_item" can be one of the following:

Profile the boundary for the trace line is the intersection point with the trace line of the referred Profile;

Panel_system the boundary for the trace line is specified by the intersection with a boundary of the Panel_system the Profile belongs to or the intersection with the (implicit or explicit given) moulded surface;

Plate if the Plate is a bracket one and belongs to the same Panel_system as the Profile does the boundary for the trace line is specified by the intersection with the (implicit or explicit given) moulded surface;

Seam the boundary for the trace line is the intersection point;

Interior_cutout if the Interior_cutout belongs to the same Panel_system as the Profile does, the boundary of the trace line is the intersection point with the contour.

The data associated with a Profile_boundary_relationship are the following:

— boundary_index.

The boundary_index specifies the index of the referred boundary in the boundary list of the Panel_system_design_definition associated with item_2 (in case item_2 is a Panel_system)

NOTE If item_2 is a Panel_system and boundary_index is not given intersection with the moulded surface of the Panel_system is taken into account.

The boundary_index need not to be specified for a particular Profile_boundary_relationship.

4.2.206 Profile_cross_section

A Profile_cross_section is a generic indicator of the standard or organisation-unique two-dimensional shape of an extruded manufactured or purchased structural part used to stiffen a plate or to provide other structural support. A Profile_cross_section may be an Explicit_profile_cross_section (see 4.2.88), a Library_profile_cross_section (see 4.2.141) or a Parametric_profile_cross_section (see 4.2.178).

EXAMPLE 1 ISO 657, ISO 1035, and ISO 9034 specify dimensions of some standard profile cross sections available from major steel manufacturers.

The data associated with a Profile_cross_section are the following:

- designation;
- section_properties.

4.2.206.1 designation

The designation specifies the part/stock designation name.

4.2.206.2 section_properties

The section_properties specify the derived analytical properties of the geometric shape of the profile cross section that determine its load-bearing capacity. See 4.3.126 for the application assertion.

4.2.207 Profile_curve_trace_line

A Profile_curve_trace_line is the "just geometry" type of trace line for a Profile (see 4.2.203).

The data associated with a Profile_curve_trace_line are the following:

- curve;
- displacement.

4.2.207.1 curve

The curve specifies the curve that defines the trace line geometry. See 4.3.127 for the application assertion.

4.2.207.2 displacement

The displacement specifies the displacement specifying the offset of the trace line orthogonal off the curve.

4.2.208 Profile_design_definition

A Profile_design_definition is a type of Structural_part_design_definition (see 4.2.271) that describes the Profile (see 4.2.203) from the detailed design point of view.

The data associated with a Profile_design_definition are the following:

- border;
- cross_section;
- cross_section_placement;
- defined_for;
- mirrored;
- trace_line;
- twist.

4.2.208.1 border

The border specifies that if the trace line of a Profile is not bounded or closed the border is to be provided by 2 Profile_boundary's; boundary can be one of

- a) an intersection point between the trace line and given line;
- b) an intersection point between the trace line and the trace line of a given Profile;
- c) an intersection point between the trace line and a boundary of the the Panel_system the Profile belongs to;
- d) an intersection point between the trace line and the moulded surface of a given Panel_system;
- e) an intersection point between the trace line and the x-y-plane of a bracket belonging to the same Panel_system;
- f) an intersection point between the trace line and a Seam;
- g) an intersection point between the trace line and the contour of a hole belonging to the same Panel_system.

The border need not to be specified for a particular Profile_design_definition. Each border may be one of the following: Line (see ISO 10303, Part 42), Profile_boundary_relationship (see

4.2.205) `Point_on_curve` (see ISO 10303, Part 42), `Cartesian_point` (see ISO 10303, Part 42). See 4.3.129, 4.3.131, 4.3.132, and 4.3.134 for the application assertions.

4.2.208.2 cross_section

The `cross_section` specifies the cross section of the profile.

4.2.208.3 cross_section_placement

The `cross_section_placement` specifies the mapping of the profile stock onto the extrusion trace, beginning at `border[1]` and ending at `border(2)`. The placement contains the extrusion starting point, axis 1 is the cross-section's local u axis, and axis 2 is the cross-section's local v axis. The cross-section's local u-v plane is everywhere normal to the extrusion curve. See 4.3.128 for the application assertion.

4.2.208.4 defined_for

The `defined_for` specifies the Profile. See 4.3.133 for the application assertion.

4.2.208.5 mirrored

The `mirrored` describes if the cross section is mirrored with respect to its definition.

4.2.208.6 trace_line

The `trace_line` specifies the trace line of the Profile (if not specified the local x-axis is implicitly assumed as trace line); trace line can be one of

- a) a curve parallel to a given curve with a given displacement;
- b) a curve parallel to a boundary of a `Panel_system` with a given displacement;
- c) a line parallel to the tangent on a boundary of the same `Panel_system` in the intersection point with a given line;
- d) a curve parallel to a hole contour of the same panel with a given displacement.

The `mirrored` need not to be specified for a particular `Profile_design_definition`. Each `trace_line` may be one of the following: `Profile_curve_trace_line` (see 4.2.207), `Profile_trace_line_relationship` (see 4.2.214), `External_instance_reference` (see 4.2.93). See 4.3.130, 4.3.136 and 4.3.137 for the application assertion.

4.2.208.7 twist

The `twist` specifies the angle of the cross-section's local v axis at locations along the extrusion trace for profiles that twist along their length. The location and orientation of twist is specified

at explicit points along the length. The change in orientation between these locations is assumed to be smooth, i.e., not knuckled. See 4.3.138 for the application assertion.

4.2.208.8 web_angle

The web_angle specifies the inclination of the profile web around to the trace curve direction

4.2.209 Profile_functional_definition

A Profile_functional_definition is a type of Structural_part_functional_definition (see 4.2.272) that defines the functional role of a Profile (see 4.2.203); the role may be a pre-defined one or may be user-defined.

The data associated with a Profile_functional_definition are the following:

- defined_for.

The defined_for specifies the redefinition of the defined_for attribute, a Profile_functional_definition is only valid for one or multiple Profile. See 4.3.139 for the application assertion.

4.2.210 Profile_manufacturing_definition

A Profile_manufacturing_definition is a type of Structural_part_manufacturing_definition (see 4.2.275) that is a production lifecycle stage view of a profile part that represents characteristics and attributes directly related to the manufacturing of the part. An earlier detailed design part stage view may or may not exist for the part. At this stage, structural part features are identified that affect aspects of the part that have to do with processes that will be used to cut and form the part. Some examples of the kinds of features identified during this stage are edge grinds, layout marks, and part stock. Information relating the structural part to the raw material stock it will be manufactured from is also added during this stage. A 2D flat pattern representation may be defined for the part, enabling it to be nested and cut from profile stock.

The data associated with a Profile_manufacturing_definition are the following:

- defined_for;
- inverse_bend_trace;
- outer_flange_contour;
- outer_web_contour.

4.2.210.1 defined_for

The defined_for specifies the redeclaration of the defined_for attribute to be allowed to point to Profile only. See 4.3.141 for the application assertion.

4.2.210.2 inverse_bend_trace

The `inverse_bend_trace` specifies the curve in the representation of a curved profile that indicates the amount that the profile is to be shaped during manufacturing. As the profile part is bent into its correct final shape, the inverse bend trace will become a straight line. See 4.3.140 for the application assertion.

4.2.210.3 outer_flange_contour

The `outer_flange_contour` specifies the curve in the representation that specifies the outer boundary of the part's flange geometry that will be burned during the manufacturing process. This is the resulting curve after the resolution of manufacturing features such as added material feature and weld_shrinkage allowance, and edge and corner features. See 4.3.140 for the application assertion.

4.2.210.4 outer_web_contour

The `outer_web_contour` specifies the curve in the representation that specifies the outer boundary of the part's web geometry that will be burned during the manufacturing process. This is the resulting curve after the resolution of manufacturing features such as added material feature and weld_shrinkage allowance, and edge and corner features. See 4.3.140 for the application assertion.

4.2.211 Profile_rejection

The `Profile_rejection` is a type of `Structural_class_rejection` (see 4.2.263) which confirms that a `Profile_design_definition` (see 4.2.208) is not approved. The reasons for rejection and existing requirements for the profile are specified.

NOTE `Profile_rejection` specifies additional information clarifying the rule requirements for the Profile. A yard has several options for action; increase the profile dimensions, improve the material quality, apply another profile etc. Although it is not mandatory, a class society may suggest improvements in these areas. improved, another stiffening may be applied etc. This may then be done using `Structural_class_rejection.proposed_alternative`.

The data associated with a `Profile_rejection` are the following:

— `subject`.

The `subject` specifies the rejected `Profile_design_definition`. See 4.3.142 for the application assertion.

4.2.212 Profile_relationship

A `Profile_relationship` is a type of `Structural_part_relationship` (see 4.2.277) that describes the relationship between a Profile (see 4.2.203) and another "structural item". "Structural item" can be one of

- a) Structural_system;
- b) Structural_part;
- c) Structural_feature.

The data associated with a Profile_relationship are the following:

- item_1.

The item_1 specifies the Profile_relationship shall relate a Profile with something else. See 4.3.143 for the application assertion.

4.2.213 Profile_renewal_definition

A Profile_renewal_definition is a type of Definition (see 4.2.57) that specifies the rule and the renewal section modulus of a Profile and its rule inertia. Rule values are values that are required by e.g. classification societies for approval. These properties are specified during the newbuilding of the ship and are used during its operation, for example during classification society surveys.

The data associated with a Profile_renewal_definition are the following:

- defined_for;
- renewal_modulus;
- rule_inertia;
- rule_section_modulus.

4.2.213.1 defined_for

The defined_for specifies the Profiles that this Definition is valid for. See 4.3.144 for the application assertion.

4.2.213.2 renewal_modulus

The renewal_modulus specifies the minimum section modulus of a Profile, below which it must be replaced or repaired. replaced or repaired.

4.2.213.3 rule_inertia

The rule_inertia specifies the minimum allowable thickness of a Profile during the life-time of the Profile. Rule_thickness is specified in the Classification Rules and Regulations.

4.2.213.4 rule_section_modulus

The `rule_section_modulus` specifies the minimum allowable section modulus of a Profile during its life-time. `Rule_section_modulus` is specified in the Classification Rules and Regulations.

4.2.214 Profile_trace_line_relationship

A `Profile_trace_line_relationship` is a type of `Profile_relationship` (see 4.2.212) that defines the trace line of a Profile by referring to a "structural item". "Structural_item" can be one of the following:

`Panel_system` the trace line is specified by a boundary or the tangent on a boundary (at the intersection point with an auxiliary line) and an orthogonal displacement off that boundary or tangent;

`Interior_cutout` the trace line is specified by the contour and an orthogonal displacement off that contour.

`Seam` the trace line is specified by the contour and an orthogonal displacement off that contour.

The data associated with a `Profile_trace_line_relationship` are the following:

- `auxiliary_line`;
- `boundary_index`;
- `displacement`.

4.2.214.1 auxiliary_line

The `auxiliary_line` specifies the point - as the intersection point of this line with the referenced boundary contour - at which the tangent on the boundary referenced by `boundary_index` is to be taken as traceline. The `auxiliary_line` need not to be specified for a particular `Profile_trace_line_relationship`. See 4.3.145 for the application assertion.

4.2.214.2 boundary_index

The `boundary_index` specifies the index of the referred boundary in the boundary list of the `Panel_system_design_definition` associated with `item_2` (in case `item_2` is a `Panel_system`) The `boundary_index` need not to be specified for a particular `Profile_trace_line_relationship`.

4.2.214.3 displacement

The `displacement` specifies a displacement specifying the offset orthogonal off the boundary referred to by `boundary_index`.

4.2.215 Property_BSU

A Property_BSU (property basic semantical unit) identifies a property of a class in a parts library.

The data associated with a Property_BSU are the following:

- name_scope;
- version.

4.2.215.1 name_scope

The name_scope specifies the class this property is belonging to, i.e. the class context. See 4.3.146 for the application assertion.

4.2.215.2 version

The version specifies the designation of the version of the information piece.

4.2.216 Property_value

A Property_value may be a Real_property_value (see 4.2.220), a Integer_property_value (see 4.2.126), a Number_property_value (see 4.2.167), a String_property_value (see 4.2.256), a Boolean_property_value (see 4.2.18), or a Logical_property_value (see 4.2.149). A Property_value provides a value for a property as specified in the property basic semantical unit. The value type is specified in subtypes.

The data associated with a Property_value are the following:

- property_identifier.

The property_identifier specifies the property this value is for, i.e. the property context. See 4.3.147 for the application assertion.

4.2.217 Proprietary_profile_cross_section

A Proprietary_profile_cross_section is a type of Parametric_profile_cross_section (see 4.2.178) that is not specified by national or international standard.

4.2.218 Ratio_measure_with_unit

A Ratio_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a ratio as defined in ISO 31 (clause 2).

4.2.219 Ratio_unit

A Ratio_unit is a type of Named_unit (see 4.2.164) that is the unit in which the relation between two physical quantities that are of the same kind is measured.

4.2.220 Real_property_value

A Real_property_value is a type of Property_value (see 4.2.216) that provides a value for a property of type REAL.

The data associated with a Real_property_value are the following:

— val.

The val specifies the value of type REAL.

4.2.221 Reason_for_decision

Reason_for_decision describes the background for a decision. This background shall be given as a reference to a document and as an informal textual description.

The data associated with a Reason_for_decision are the following:

— annotations;

— rule_reference.

4.2.221.1 annotations

The annotations specifies the informative textual description of the reason of a decision.

4.2.221.2 rule_reference

The rule_reference specifies the reason for a decision in terms of a reference to a document. See 4.3.148 for the application assertion.

4.2.222 Rectangular_cutback_corner_design_definition

A Rectangular_cutback_corner_design_definition is a type of Corner_cutout_design_definition (see 4.2.45) that specifies the removal of material from a corner such that the resulting cutoff edges are parallel to the original corner edges. The local origin is placed at the corner that shall be removed and the two (three) axis points along the edges.

The data associated with a Rectangular_cutback_corner_design_definition are the following:

— x_depth;

- `y_depth`;
- `radius`.

4.2.222.1 `x_depth`

The `x_depth` specifies the distance from local origin along x-axis.

4.2.222.2 `y_depth`

The `y_depth` specifies the distance from local origin along y-axis.

4.2.222.3 `radius`

The `radius` specifies the radius of the circular arc of the new corner inside the cutback.

4.2.223 `Rectangular_cutout_design_definition`

A `Rectangular_cutout` is a type of `Interior_cutout_design_definition` (see 4.2.128) that is an opening where the opposing sides are parallel and adjacent sides are perpendicular.

NOTE Combination of `Round_corner_rectangular_cutout_design_definition` with `Round_edge_rectangular_cutout_design_definition` allowed.

The data associated with a `Rectangular_cutout_design_definition` are the following:

- `length_of`;
- `width`.

4.2.223.1 `length_of`

The `length_of` specifies the opening along y-axis.

4.2.223.2 `width`

The `width` specifies the opening along x-axis.

See 25 for a `Rectangular_cutback_corner_cutout`.

4.2.224 `Research_Ship`

A `Research_Ship` is a type of `ShipType` (see 4.2.243) that is a ship which is specially designed for research.

The data associated with a `Research_Ship` are the following:

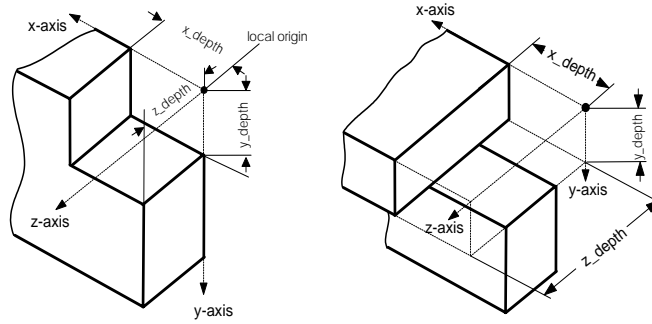


Figure 25 – Rectangular_cutback_corner_cutout

— has_type.

The has_type specifies the type of research ships.

The value of the has_type shall be one of the following:

— user defined

4.2.224.1

user defined

can be any ship type which is not in the enumeration list. Details should be found in the ship_type_description

4.2.225 Regulation

A Regulation specifies the set of all international and national regulations as well as standards which apply to the ship.

The data associated with a Regulation are the following:

— international_regulations;

— national_regulations;

— standards.

4.2.225.1 international_regulations

The `international_regulations` specifies all relevant international regulations which apply to the ship. See 4.3.149 for the application assertion.

4.2.225.2 national_regulations

The `national_regulations` specifies all relevant national regulations which apply to the ship. See 4.3.149 for the application assertion.

4.2.225.3 standards

The `standards` specifies all relevant standards which apply to the ship. See 4.3.149 for the application assertion.

4.2.226 Revision

A Revision is a type of `Versionable_object` (see 4.2.307) that is the link between the thing of interest, the context, (e.g. a HCS) and the Definitions of its constituents, the members, (e.g. its plate definitions, but only those that belong to the same version). A Revision is not created automatically, but has to be created explicitly each time it is needed.

The data associated with a Revision are the following:

- members;
- name;
- reason.

4.2.226.1 members

The `members` specifies the `Versionable_objects` the Revision consists of. See 4.3.150 for the application assertion.

4.2.226.2 name

The `name` specifies the Label which identifies a particular revision.

4.2.226.3 reason

The `reason` specifies the description of what caused the creation of a new revision.

4.2.227 Revision_with_context

A `Revision_with_context` is a type of `Revision` (see 4.2.226) that is the link between the thing of interest, the context, (e.g. a HCS) and the Definitions of its constituents, the members, (e.g. its plate definitions, but only those that belong to the same version). The data associated with a `Revision_with_context` are the following:

— `context_of_revision`.

The `context_of_revision` specifies the link to a higher level thing of interest. This may be not always known explicitly. See 4.3.151 for the application assertion.

4.2.228 Round_bar_cross_section

A `Round_bar_cross_section` is a type of a `Bar_profile_cross_section` (see 4.2.15) identified by a solid circular shape.

NOTE ISO 1035-1 specifies dimensions for some Round bars.

The data associated with a `Round_bar_cross_section` are the following:

— `diameter`.

The `diameter` specifies the the outside diameter of the bar.

4.2.229 Round_corner_design_definition

A `Round_corner_design_definition` is a type of `Corner_cutout_design_definition` (see 4.2.45) specifies the circular arc (cylindrical surface) between two adjacent edges (surfaces).

The data associated with a `Round_corner_design_definition` are the following:

— `radius`.

The `radius` specifies the radius of the circular arc (cylindrical surface) between two adjacent edges (surfaces) of the corner.

4.2.230 Round_corner_rectangular_cutout_design_definition

A `Round_corner_rectangular_cutout` is a type of `Rectangular_cutout_design_definition` (see 4.2.223) that is an opening where the opposing sides are parallel and adjacent sides are perpendicular. The corners are rounded off with the same radius.

The data associated with a `Round_corner_rectangular_cutout_design_definition` are the following:

— `corner_radius`.

The `corner_radius` specifies the radius of the rounded corners; the same radius applies to all corners. Figure 26 shows a `Round_corner_rectangular_cutout`.

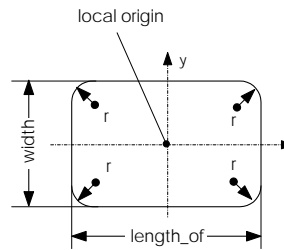


Figure 26 – Round_corner_rectangular_cutout

4.2.231 Round_edge_rectangular_cutout_design_definition

A `Round_edge_rectangular_cutout` is a type of `Rectangular_cutout_design_definition` (see 4.2.223) that has the same basic shape as a `Rectangular_cutout` but in addition two of the opposing sides are substituted by outward bended circle arcs.

The data associated with a `Round_edge_rectangular_cutout_design_definition` are the following:

- `distance`;
- `edge_radius`.

4.2.231.1 distance

The `distance` specifies the distance along x-axis of the circle center from the origin.

4.2.231.2 edge_radius

The `edge_radius` specifies the radius of the circular arc that describes the outward bended sides of the bended sides of the opening.

4.2.232 Seam

A `Seam` is a type of `Position_feature` (see 4.2.195) that provides a reference for an inner `Plate` (see 4.2.184) boundary on a `Panel_system` (see 4.2.170). Seams are usually created in the preliminary design stage in order to define where the `Panel_system` will be splitted into `Plates` in a later design stage.

The data associated with a `Seam` are the following:

— parent.

The parent specifies the Seam shall be related to a Panel_system only. See 4.3.152 for the application assertion.

4.2.233 Seam_curve_relationship

A Seam_curve_relationship is a type of Position_feature_relationship (see 4.2.197) that defines the curve of a Seam by referring to a "structural item". "Structural_item" can be one of the following:

Panel_system (see 4.2.170) the trace line is specified by a boundary of the Panel_system and an orthogonal displacement off that boundary;

Seam (see 4.2.232) the trace line is specified by the curve of another Seam and an orthogonal displacement off that curve.

The data associated with a Seam_curve_relationship are the following:

— boundary_index;

— displacement.

4.2.233.1 boundary_index

The boundary_index specifies the index of the referred boundary in the boundary list of the Panel_system_design_definition associated with item_2 (in case item_2 is a Panel_system). The boundary_index need not to be specified for a particular Seam_curve_relationship.

4.2.233.2 displacement

The displacement specifies the displacement specifying the offset off the referred geometry.

4.2.234 Seam_design_definition

A Seam_design_definition is a type of Feature_design_definition (see 4.2.96) that describes the definition of a Seam (see 4.2.232) from the view point of design.

The data associated with a Seam_design_definition are the following:

— border;

— defined_for;

— seam_curve.

4.2.234.1 border

The border specifies the Seam may optionally be bounded on one or both ends by another Seam. The border need not to be specified for a particular Seam_design_definition.

4.2.234.2 defined_for

The defined_for specifies the Seam_design_definition shall only be valid for Seams. See 4.3.153 for the application assertion.

4.2.234.3 seam_curve

The seam_curve is optional and, if present, specifies the curve defining the Seam (if not specified the local x-axis is implicitly assumed as the seam defining curve); seam curve can be one of a given curve, a curve parallel to a boundary of a Panel_system with a given displacement or a curve parallel to a given Seam with a given displacement; the seam curve is on both ends required to intersect with the parent Panel_system's boundary (as provided by the related Panel_system_design_definition) or a border must explicitly be given for the Seam on that curve end.

4.2.235 Section_modulus_measure_with_unit

A Section_modulus_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is the moment of resistance.

4.2.236 Section_modulus_unit

A Section_modulus_unit is a type of Derived_unit (see 4.2.61) that is the unit of a shape parameter that indicates the exposure of a cross section to internal stress as a result of bending. This physical quantity is the value of an area moment of inertia divided by the distance to the outermost geometry.

4.2.237 Section_properties

The Section_properties hold some basic quantities characterising the cross section of a profile which are often needed for the dimensioning and engineering calculations.

NOTE The following quantities can be derived from the attributes of Section_properties using the given formulas:

- section modulus in u ($\text{moi_u} / \text{na_u}$);
- section modulus in v ($\text{moi_v} / \text{na_v}$);
- radius of gyration in u ($\text{SQRT}(\text{moi_u} / \text{area})$);
- radius of gyration in v ($\text{SQRT}(\text{moi_v} / \text{area})$).

The data associated with the Section_properties are the following:

- area;
- na_u;
- na_v;
- moi_u;
- moi_v;
- moi_uv;
- nominalwt_per_len;
- tr.

4.2.237.1 area

The area specifies the area of solid material in the cross-section.

4.2.237.2 na_u

The na_u specifies the neutral axis in U - distance along u to centroid from local_coordinate system.

4.2.237.3 na_v

The na_v specifies the neutral Axis in V - distance along u to centroid from local_coordinate system.

4.2.237.4 moi_u

The moi_u specifies the moment of inertia in U.

4.2.237.5 moi_v

The moi_v specifies the moment of inertia in V.

4.2.237.6 moi_uv

The moi_uv specifies the cross moment of interia - zero if symmetric in U or V.

4.2.237.7 nominalwt_per_len

The nominalwt_per_len should be a derived_unit of weight per length measure. nominal weight per unit length - based on density of steel.

4.2.237.8 tr

The tr specifies the torsional rigidity.

4.2.238 Shear_bevel_design_definition

A Shear_bevel_design_definition is a type of Bevel_design_definition (see 4.2.16) that specifies the 3-dimensional surface obtained when intersecting (parts of) a corner with a plane not being parallel to any of the corner surfaces. The data associated with a Shear_bevel_design_definition are the following:

— x_z_angle.

The x_z_angle specifies the angle between bevelled side and x-axis in the x-z-plane, see figure 27).

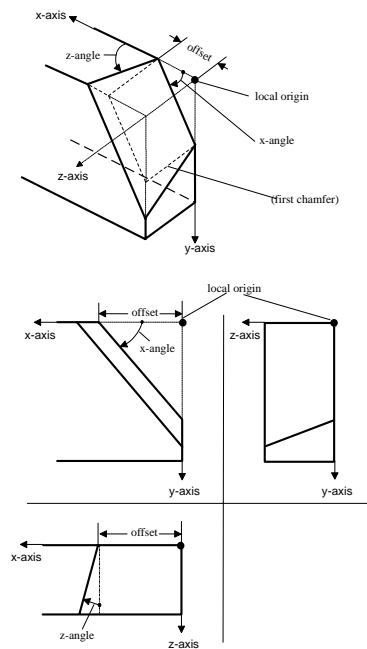


Figure 27 – Shear_bevel_corner_cutout

4.2.239 Shear_force

A Shear_force is a type of Design_load (see 4.2.64) that is one way to represent Design_loads, which is by the physical quantity force.

4.2.240 Ship

A Ship is a type of Item (see 4.2.132) that is the naval architectural object in concern. All data defining the product shall be related to a ship, which might exist in any life cycle stage. A project, which represents a ship in the early design phase, for example before contract, is also regarded as a ship.

The data associated with a Ship are the following:

- units.

The units specifies the reference to a set of pre-defined units for all types of measures that may appear in the ship model.

NOTE The name of the ship is specified as a Definition in Ship_designation, where it may be versioned.

4.2.241 Ship_designation

A Ship_designation is a type of General_characteristics_definition (see 4.2.113) that specifies the identification given to the ship in order that it can be categorized by any shipping related organization.

The data associated with a Ship_designation are the following:

- call_sign;
- flag_state;
- port_of_registration;
- ship_name;
- ship_type_description.

4.2.241.1 call_sign

The call_sign specifies the unique lifecycle identifier assigned to the ship by the flag_state for radio communication.

4.2.241.2 flag_state

The `flag_state` specifies the national authority with whom the ship is registered.

4.2.241.3 port_of_registration

The `port_of_registration` specifies the national home port of the ship. The `port_of_registration` lies within the jurisdiction of the `flag_state`.

4.2.241.4 ship_name

The `ship_name` specifies the owner assigned name of the ship.

4.2.241.5 ship_type_description

The `ship_type_description` gives more details about the function of the ship and additional information about the cargo carried. If the `ship_type` is `user_defined`, then the `ship_type_description` delivers the information for the type of the ship.

4.2.242 Ship_material_property

A `Ship_material_property` is a type of `Definition` (see 4.2.57) that is the supertype for all materials used to specify the raw materials by physical properties as a definition for Items being made of a material.

The data associated with a `Ship_material_property` are the following:

- `defined_for`;
- `density`;
- `description`;
- `material_reference`.

4.2.242.1 defined_for

The `defined_for` specifies that the `Ship_material_property` is restricted to be defined for an Item. See 4.3.155 for the application assertion.

4.2.242.2 density

The `density` specifies the density of the material.

4.2.242.3 description

The description specifies the 'name' used for that material. The description need not to be specified for a particular Ship_material_property.

4.2.242.4 material_reference

The material_reference specifies the any reference to a source where more specific information about a material is available. The material_reference need not to be specified for a particular Ship_material_property. See 4.3.154 for the application assertion.

4.2.243 ShipType

A ShipType is a type of Functional_definition (see 4.2.112) that is the description of the function, purpose or mission a ship is designed for. A ShipType is either a Carrier (see 4.2.24), a Navy_Ship (see 4.2.165), a Research_Ship (see 4.2.224), or a Working_Ship (see 4.2.326). A ship may have multiple functions. In this case several Entities of this type shall be defined and assigned to an instance of Ship. It is almost impossible to provide a complete list of ship types. Therefore only the major types are named explicitly, other may be assined via the description attribute.

The data associated with a ShipType are the following:

- defined_for;
- description.

4.2.243.1 defined_for

The defined_for specifies the ship for which the ship type is defined. See 4.3.156 for the application assertion.

4.2.243.2 description

The description attribute is used to assign a ship type, which is not covered in the list or needs further explanation.

4.2.244 Shipyard_designation

A Shipyard_designation is a type of General_characteristics_definition (see 4.2.113) that specifies the identification given to the ship by the shipbuilder.

The data associated with a Shipyard_designation are the following:

- role;
- shipyard;

- shipyard_new_building_id;
- shipyard_project_name.

4.2.244.1 role

The role specifies the contractual obligation the shipyard has in relation to the ship.

The value of the role shall be one of the following:

- prime;
- prime build;
- prime design;
- prime repair;
- subcontractor.

NOTE See 4.2.244.1.1 - 4.2.244.1.4 for the definition of each allowable value for role.

4.2.244.1.1

prime

the prime contractor for the ship

4.2.244.1.2

prime build

the prime contractor, with contract responsibilities for manufacture of the ship

4.2.244.1.3

prime design

the prime contractor, with contract responsibility for the design of the ship

4.2.244.1.4

prime repair

the prime contractor, with contract responsibilities for repair of the ship

4.2.244.1.5

subcontractor

a subcontractor for the ship

4.2.244.2 shipyard

The shipyard specifies the name and organizational details of the shipyard. See 4.3.157 for the application assertion.

4.2.244.3 shipyard_new_building_id

The shipyard_new_building_id specifies the identifier for the ship that is assigned by the shipyard after an order has been confirmed. The shipyard_new_building_id need not to be specified for a particular Shipyard_designation.

4.2.244.4 shipyard_project_name

The shipyard_project_name is an identifier for the ship that is assigned by the shipyard on receipt of an order, or tender, for a new ship.

4.2.245 Si_unit

A Si_unit is the fixed quantity used as a standard in terms of which items are measured as defined by ISO 1000 (clause 2).

The data associated with a Si_unit are the following:

- dimensions;
- name;
- prefix.

4.2.245.1 dimensions

The dimensions is the SI dimensions which is derived from the Named_unit. See 4.3.158 for the application assertion.

4.2.245.2 name

The name is the word or group of words by which the is referred to.

4.2.245.3 prefix

The prefix is the SI prefix.

4.2.246 Spacing_position

A Spacing_position is either a Longitudinal_position (see 4.2.150), a Transversal_position (see 4.2.297), a Vertical_position (see 4.2.309), or a Spacing_position_with_offset (see 4.2.247) that is a position on one of the global co-ordinate axes of the ship (see figure 28) that is used as a reference point for any geometrical or structural item during the design and manufacture of the ship.

The data associated with a Spacing_position are the following:

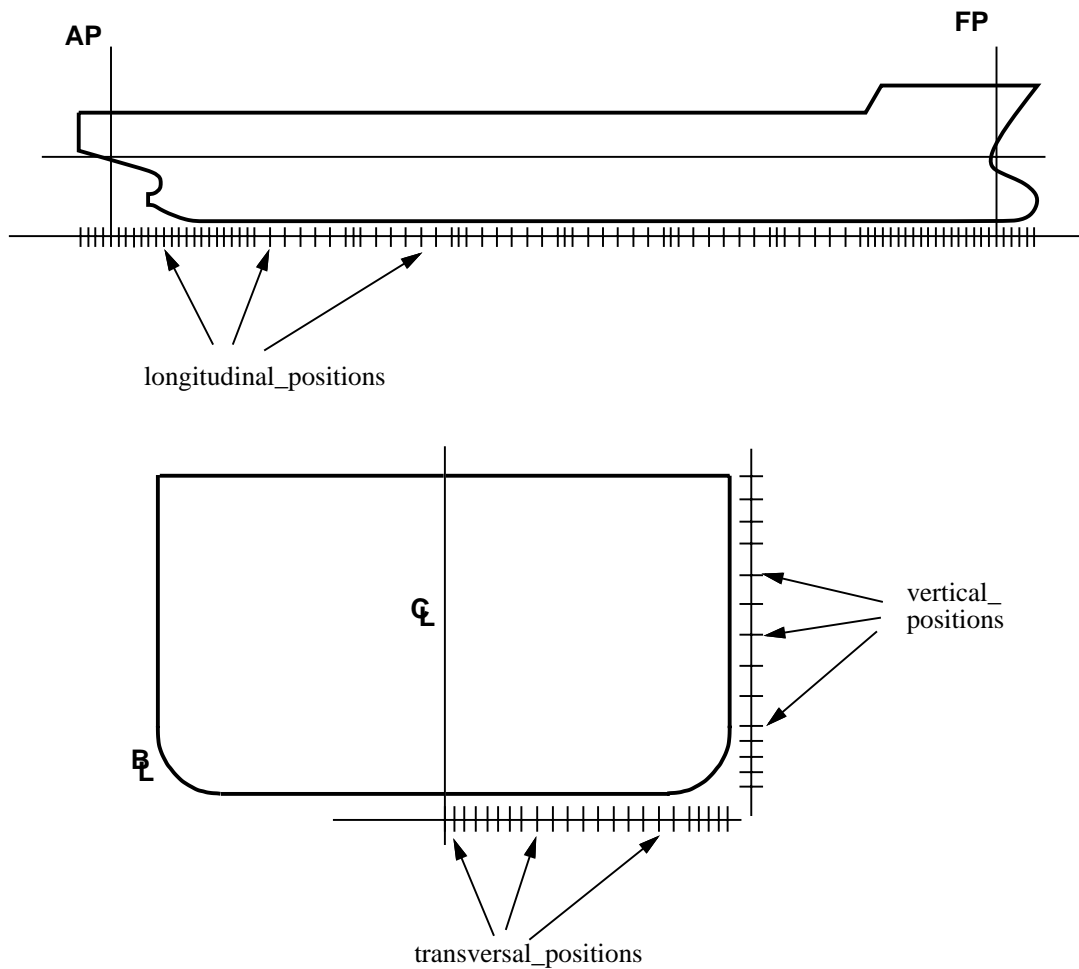


Figure 28 – Spacing positions

- id;
- position.

4.2.246.1 id

The id specifies the unique identification given to the Spacing-position.

EXAMPLE 1 Typically these are specified by LFR123, TFR10, 100, 100.1, A.

4.2.246.2 position

The position specifies the distance to the origin of the ships global co-ordinate system. The axis, where the distance is measured depends on the type of Spacing-position.

4.2.247 Spacing_position_with_offset

A `Spacing_position_with_offset` is a type of `Spacing_position` (see 4.2.246) that is a position defined by an offset to an existing spacing position on one of the global co-ordinate axes of the ship. It is used as a reference point for any geometrical or structural item during the design and manufacture of the ship.

The data associated with a `Spacing_position_with_offset` are the following:

- `offset`;
- `position`;
- `relating_spacing_position`.

4.2.247.1 offset

The `offset` specifies the distance to the relating spacing position. The axis on which the distance is measured depends on the type of the relating `Spacing_position`.

4.2.247.2 position

The `position` specifies the distance to the origin of the ship's global co-ordinate system. The axis on which the distance is measured depends on the type of the relating `Spacing_position`.

4.2.247.3 relating_spacing_position

The `relating_spacing_position` specifies the spacing position from where the offset is taken to identify the `Spacing_position_with_offset`. See 4.3.159 for the application assertion.

4.2.248 Spacing_table

A `Spacing_table` is a type of `Definition` (see 4.2.57) that is a collection of `Spacing_positions` (see 4.2.246) which defines a list of reference points along one of the co-ordinate axes of the Ship. There are several specific types of the `Spacing_table`, which can be chosen.

EXAMPLE 1 A frame table is a type of `Spacing_table`, where the frame positions would be longitudinal positions.

The data associated with a `Spacing_table` are the following:

- `description`;
- `name`;
- `spacing_table_representations`.

4.2.248.1 description

The description specifies the textual account of the reason why the Spacing_table was created and any additional text that is required to describe the purpose of the Spacing_table. The description need not to be specified for a particular Spacing_table.

4.2.248.2 name

The name specifies the context specific identification for the Spacing_table. The name need not to be specified for a particular Spacing_table.

4.2.248.3 spacing_table_representations

The spacing_table_representations specifies the positions which make up the table on the co-ordinate axis that are of interest. See 4.3.160 for the application assertion.

4.2.249 Speed_measure_with_unit

A Speed_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is speed.

4.2.250 Speed_unit

A Speed_unit is a type of Derived_unit (see 4.2.61) that is the unit of the physical quantity of speed, which is distance made good over time.

4.2.251 Spot_seam_weld

A Spot_seam_weld is a type of Welded_joint_design_definition (see 4.2.324) where the weld performed by locally heating and simultaneously pressurizing through the electrode, with current and welding force concentrated in comparatively small area (spot or seam) of assembly formed by holding the lapped base metal with a properly formed electrode trip.

The data associated with a Spot_seam_weld are the following:

— weld_context.

The weld_context specifies the indicator as to whether a spot/seam weld represents a spot weld or a seam weld.

Figure 29 shows spot and plug weld.

4.2.252 Square_tube_cross_section

A Square_tube_cross_section is a type of Non_circular_hollow_profile_cross_section (see 4.2.166) that is square or rectangular in shape and has a non-solid core.

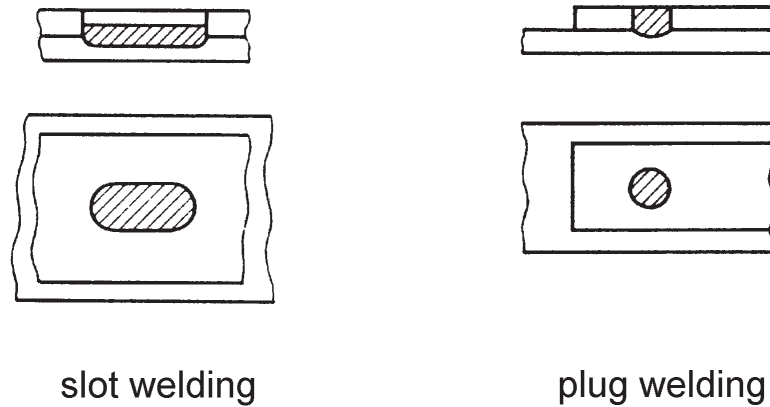


Figure 29 – Spot and plug weld

The data associated with a `Square_tube_cross_section` are the following:

- `depth`;
- `wall_thk`;
- `width`.

4.2.252.1 depth

The `width` specifies the depth of the tube.

4.2.252.2 wall_thk

The `wall_thk` specifies the wall thickness of the `Square_tube_cross_section`.

4.2.252.3 width

The `width` specifies the width of the tube.

4.2.253 Station_table

A `Station_table` is a type of `Transversal_table` (see 4.2.298) whose positions are a reference for the location of stations and are located on the global x axis.

NOTE Stations are used in the design process of a ship and the station curves are curves of transversal sections through the ship hull. There are usually 20 stations, but the number can differ from shipyard to shipyard.

The data associated with a `Station_table` are the following:

- `spacing_table_representations`.

The `spacing_table_representations` specifies the longitudinal positions of the stations.

4.2.254 Stress_measure_with_unit

A `Stress_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is stress within a material.

4.2.255 Stress_unit

A `Stress_unit` is a type of `Derived_unit` (see 4.2.61) that is the unit of the physical quantity of stress, which is force per area.

4.2.256 String_property_value

A `String_property_value` is a type of `Property_value` (see 4.2.216) that provides a value for a property of type `STRING`.

The data associated with a `String_property_value` are the following:

- `val`.

The `val` specifies the value of type `STRING`.

4.2.257 Structural_added_material_boundary_relationship

A `Structural_added_material_boundary_relationship` is a type of `Structural_feature_relationship` (see 4.2.267) that specifies the topological relationship between a `Structural_added_material_feature` and the boundary of a `Plate`, `Profile`, or `Corrugated_part`.

The data associated with a `Structural_added_material_boundary_relationship` are the following:

- `boundary_index`;
- `item_1`.

4.2.257.1 boundary_index

The `boundary_index` specifies the index of the boundary in the boundary list of the `Profile_design_definition`, `Plate_design_definition`, or `Corrugated_part_design_definition`, depending on

which one is referenced by item_2, that is to have material added to its edge for manufacturing.

4.2.257.2 item_1

The item_1 specifies the relating item of the relationship shall be a Structural_added_material_feature. See 4.3.161 for the application assertion.

4.2.258 Structural_added_material_feature

A Structural_added_material_feature is a type of Structural_manufacturing_feature (see 4.2.268) that is one type of part Feature that represents characteristics that describe the addition of material to a structural part edge to provide margins in joining the parts at later manufacturing stages and maintaining critical dimensions. It is also referred to as green material. Some of this added material may be trimmed off when the parts are joined. An added material feature will modify the topological representation of the part. The edge it is applied to will be superseded by a new set of edges that represent the "extended" part's boundary.

4.2.259 Structural_added_material_feature_design_definition

A Structural_added_material_feature_design_definition is a type of Feature_design_definition (see 4.2.96) that is one type of part Feature that represents characteristics that describe the addition of material to a structural part edge to provide margins in joining the parts at later manufacturing stages and maintaining critical dimensions. It is also referred to as green material. Some of this added material may be trimmed off when the parts are later joined. An added material feature will modify the topological representation of the part. The edge it is applied to will be superseded by a new set of edges that represent the "extended" part's boundary.

The data associated with a Structural_added_material_feature_design_definition are the following:

- added_material_length;
- defined_for.

4.2.259.1 added_material_length

The added_material_length specifies the measure of the amount of material that is to be added to a part edge. This dimension is measured normal to the part's edge on the moulded surface.

4.2.259.2 defined_for

The defined_for specifies the Structural_added_material_feature that the Structural_added_material_feature_design_definition is defined for. See 4.3.162 for the application assertion.

4.2.260 Structural_class_approval

A `Structural_class_approval` is a type of `Structural_class_approval_result` (see 4.2.261) that specifies a positive result of the assessment of a `Structural_part` (see 4.2.269) against for example the rules of a classification society; the part did pass the verification.

4.2.261 Structural_class_approval_result

A `Structural_class_approval_result` specifies the result of the assessment of a `Structural_part` (see 4.2.269) against for example the rules of a classification society. Subtypes of this entity indicate whether the part passed the verification or not. A `Structural_class_approval_result` is either a `Structural_class_approval` (see 4.2.260), a `Structural_class_rejection` (see 4.2.263), or a `Structural_class_conditional_approval` (see 4.2.262).

The data associated with a `Structural_class_approval_result` are the following:

— explanations.

The explanations specifies the reasons that led to the decision for approval respectively disapproval. There may not always be a reason specified, especially not in the case of approval. See 4.3.163 for the application assertion.

4.2.262 Structural_class_conditional_approval

A `Structural_class_conditional_approval` is a type of `Structural_class_approval_result` (see 4.2.261) that specifies a mainly positive result of the assessment of a `Structural_part` (see 4.2.269) against the rules of for example a classification society. The part did pass the verification under the condition that clearly specified changes will be incorporated.

The data associated with a `Structural_class_conditional_approval` are the following:

— explanations.

The explanations specifies The condition under which the approval is valid. At least one condition shall be given. See 4.3.164 for the application assertion.

4.2.263 Structural_class_rejection

`Structural_class_rejection` is a type of `Structural_class_approval_result` (see 4.2.261) that specifies a negative result of the assessment of a `Structural_part` (see 4.2.269) against for example the rules of a classification society; the part did not pass verification. The rejection may be accompanied by an alternative design for the investigated part.

The data associated with a `Structural_class_rejection` are the following:

— explanations;

— proposed_alternative.

4.2.263.1 explanations

The explanations specifies the reason for the disapproval. At least one reason shall be given. See 4.3.166 for the application assertion.

4.2.263.2 proposed_alternative

The proposed_alternative specifies the collection of alternative Definitions that would pass the assessment. The collection may include different alternatives, or it may contain one alternative only the properties of which are spread across several Definitions. See 4.3.165 for the application assertion.

4.2.264 Structural_cutout

A Structural_cutout is a type of Structural_feature (see 4.2.266) that specifies the nature of cutouts, that is the removal of material from a Structural_part (see 4.2.269) or a Structural_system (see 4.2.279).

NOTE Cutouts are distinguished by the number of part boundaries that they affect into: - Interior_cutouts, no boundary crossed - Edge_cutouts, 1 boundary crossed - Corner_cutouts, 2 boundaries crossed. Also Bevels, such as used for weld preparation are Structural_cutouts; they modify a boundary of a Structural_part.

The global position of a Structural_cutout is given by its parent item, e.g. by a Plate. If a Structural_cutout shall not be at the origin of its parent, it needs to be located relatively to its parent by using Local_co_ordinate_system. Complex cutouts may be constructed using the concept of composite features.

EXAMPLE 1 Cutouts may be represented as the result of profile penetrations of plates, for rat-holes and so on.

Figure 30 shows plate cutouts. Figure 31 shows profile cutouts.

4.2.265 Structural_cutout_boundary_relationship

A Structural_cutout_boundary_relationship is a type of Structural_feature_relationship (see 4.2.267) that specifies the topological relationship between a Structural_cutout (see 4.2.264) and the boundary of a Plate (see 4.2.184) or Panel_system (see 4.2.170).

NOTE Structural_cutout_boundary_relationships may be instantiated as Corner_cutout_boundary_relationships or Edge_cutout_boundary_relationships.

The data associated with a Structural_cutout_boundary_relationship are the following:

— boundary_index;

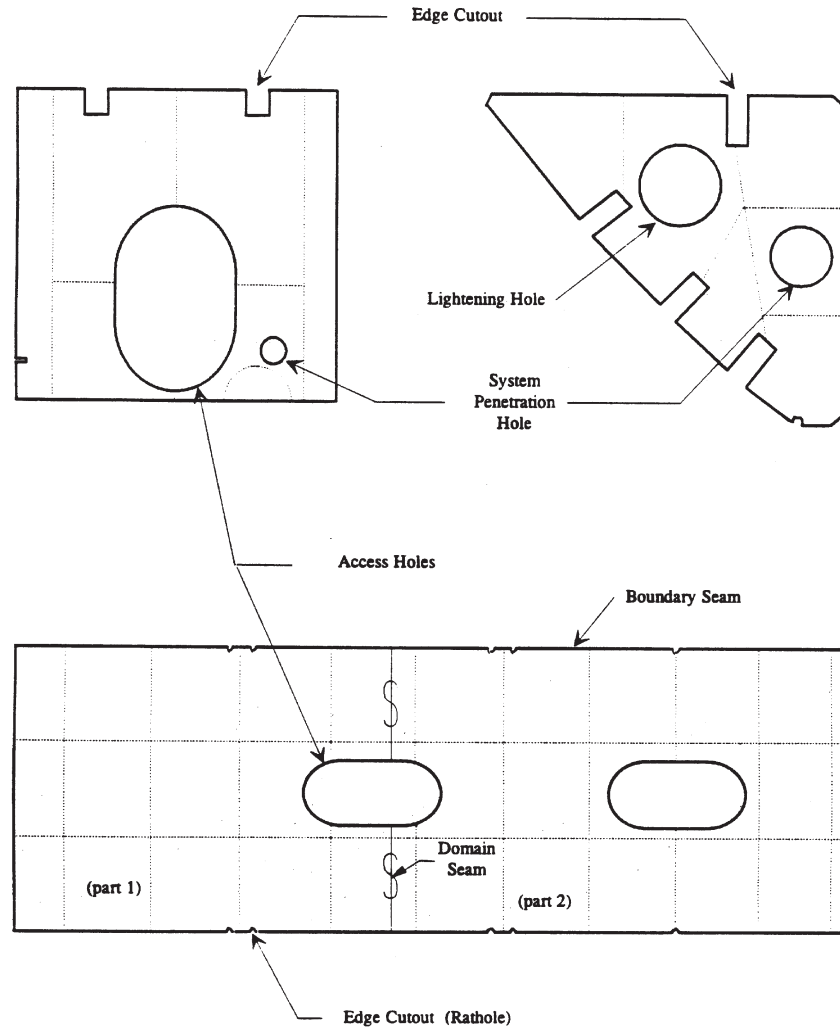
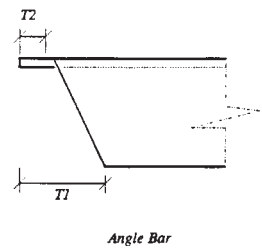
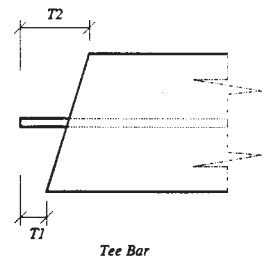
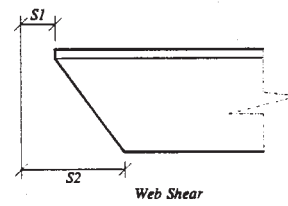
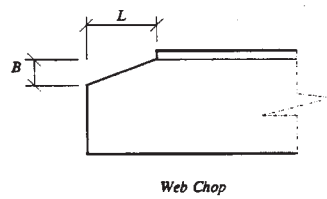
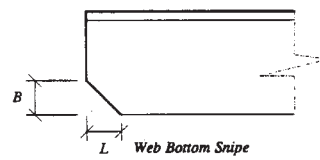
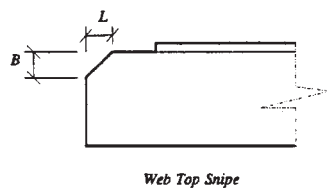
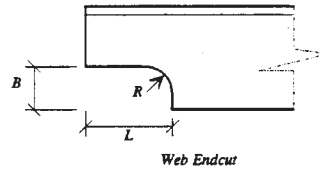
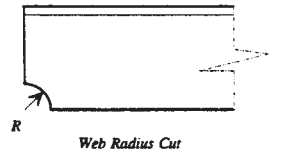
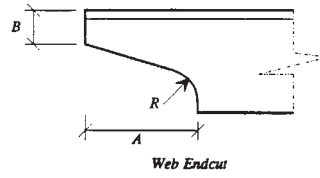
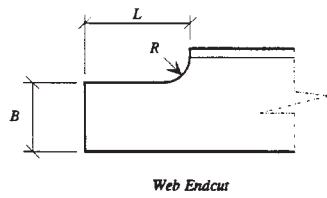


Figure 30 – Plate cutouts

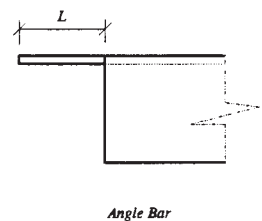
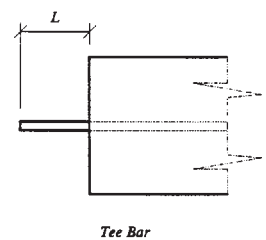
— item_1.

4.2.265.1 boundary_index

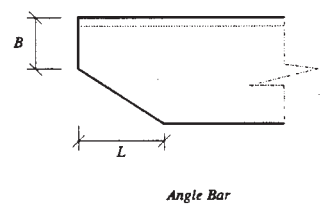
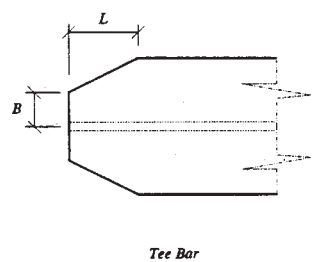
The `boundary_index` specifies the index of the referred boundary in the boundary list of the `Panel_system_design_definition` or `Plate_design_definition`, depending on which one is referenced by `item_2`.



Flange Shear



Flange Cutback



Flange Snipe

4.2.265.2 item_1

The item_1 specifies the relating item of the relationship shall be a Structural_cutout. See 4.3.167 for the application assertion.

4.2.266 Structural_feature

A Structural_feature is a type of Feature (see 4.2.95) that specifies a Feature that can only be applied to Structural_parts or Structural_systems. If a Structural_feature shall be a Composite_feature it shall be instantiated as a complex subtype of Feature.

The data associated with a Structural_feature are the following:

— parent.

The parent describes the element of the ship structure that the Structural_feature is applied to. See 4.3.168 for the application assertion.

NOTE Also Systems are Items and may, therefore, be parents to Structural_features.

4.2.267 Structural_feature_relationship

A Structural_feature_relationship is a type of Feature_relationship (see 4.2.97) that defines the association of a Structural_feature (see 4.2.266) with another Item.

The data associated with a Structural_feature_relationship are the following:

— item_1.

The item_1 specifies the relating item of the relationship shall be a Structural_feature. See 4.3.169 for the application assertion.

4.2.268 Structural_manufacturing_feature

A Structural_manufacturing_feature is a type of Structural_feature (see 4.2.266) that can be applied to Structural_parts to specify production engineering information or constraints that are to be applied to the design definition of the Structural_part for the purposes of manufacturing or assembly of the part.

The data associated with a Structural_manufacturing_feature are the following:

— parent.

The parent describes the element of the ship structure that the Structural_manufacturing_feature is applied to. See 4.3.170 for the application assertion.

4.2.269 Structural_part

A `Structural_part` is a type of `Part` (see 4.2.179) that is an atomic (in terms of ship structure design) piece of material that is (to be) fabricated from stock material mainly by cutting (and maybe in addition bending) processes but not by aggregation processes (such as `Plates` and `Profiles`). A `Structural_part` may be part of different product model associations such as `Assemblies`, topological relationships, `Joints` etc.

4.2.270 Structural_part_connection_implementation

A `Structural_part_connection_implementation` is a type of `Item` (see 4.2.132) that contains the information requirements necessary to present the connectivity between structural parts. Two or more parts are joined together, these requirements are fulfilled by structural parts connection implementation which represents a physical realization of the connection.

The data associated with a `Structural_part_connection_implementation` are the following:

- `realization_of`.

The `realization_of` specifies the methods which represents a physical realization of the connection. For example, the weld is a widely used joint method for the shipbuilding.

4.2.271 Structural_part_design_definition

A `Structural_part_design_definition` is a type of `Design_definition` (see 4.2.63) that describes the abstract definition concept of any `Structural_part` (see 4.2.269) from the view of design.

The data associated with a `Structural_part_design_definition` are the following:

- `defined_for`;
- `mass`;
- `representations`.

4.2.271.1 defined_for

The `defined_for` specifies that the `Structural_part_design_definition` shall only be valid for a `Structural_part`. See 4.3.172 for the application assertion.

4.2.271.2 mass

The `mass` is optional and, if present, specifies the mass of the `Structural_part`.

4.2.271.3 representations

The representations specifies that the Structural_part_design_definition shall be represented by Shape_representations. See 4.3.171 for the application assertion.

4.2.272 Structural_part_functional_definition

A Structural_part_functional_definition is a type of Functional_definition (see 4.2.112) that provides the capability to assign a function to a Structural_part (see 4.2.269) such as Plate or Profile.

Figure 32 shows a deck composed from Plates and Profiles.

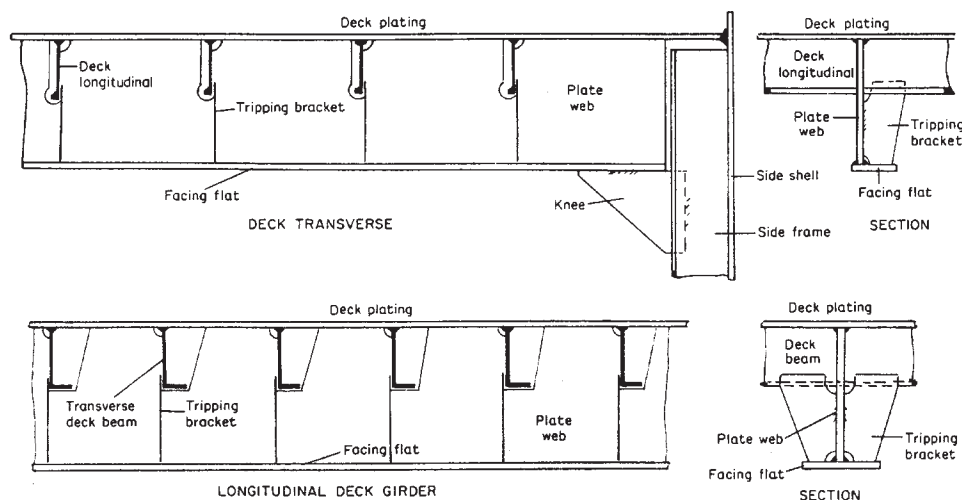


Figure 32 – A deck

Figure 33 shows a plain bulkhead composed from Plates and Profiles.

The data associated with a Structural_part_functional_definition are the following:

- defined_for;
- function.

4.2.272.1 defined_for

The defined_for specifies the redefinition of the defined_for attribute, a Structural_part_functional_definition is only valid for one or multiple Structural_parts. See 4.3.173 for the application assertion.

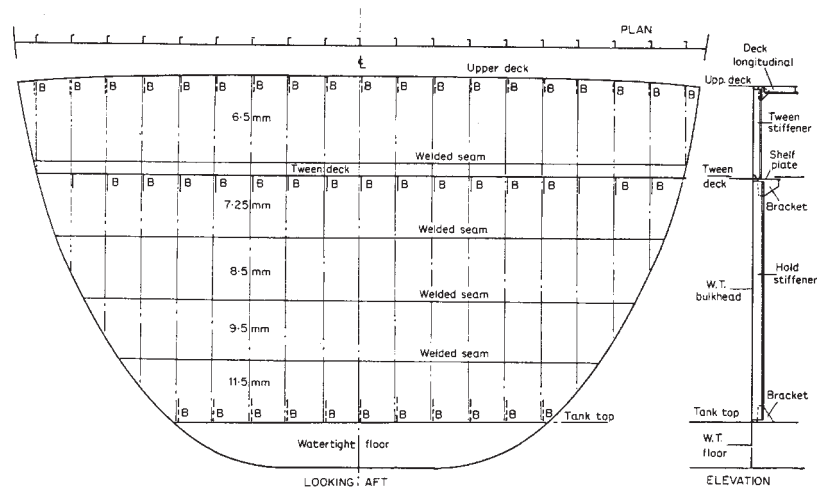


Figure 33 – A bulkhead

4.2.272.2 function

The function specifies the functionality of the Structural_part SELF points to.

The value of the function shall be one of the following:

- BEAM;
- BRACKET;
- CARLING;
- CLIP;
- DECK BEAM;
- DOUBLING PLATE;
- FACE PLATE;
- FLANGE;
- FRAME;
- HOLD PILLAR;
- LONGITUDINAL STIFFENER;
- LUG;

- PILLAR;
- SHEDDING PLATE;
- STANDARD PLATE;
- STIFFENER;
- TRANSVERSAL STIFFENER;
- USER DEFINE.
- WATERTIGHT CLIP;
- WEB;

NOTE See 4.2.272.2.1 - 4.2.272.2.20 for the definition of each allowable value for function.

4.2.272.2.1

BEAM

a beam in general

4.2.272.2.2

BRACKET

a bracket

4.2.272.2.3

CARLING

a carling

4.2.272.2.4

CLIP

a clip in general

4.2.272.2.5

DECK BEAM

a deck beam

4.2.272.2.6

DOUBLING PLATE

a doubling plate

4.2.272.2.7

FACE PLATE

a face plate

4.2.272.2.8

FLANGE

a flange

4.2.272.2.9

FRAME

a frame

4.2.272.2.10

HOLD PILLAR

a hold pillar

4.2.272.2.11

LONGITUDINAL STIFFENER

a longitudinal stiffener

4.2.272.2.12

LUG

a lug

4.2.272.2.13

PILLAR

a pillar in general

4.2.272.2.14

SHEDDING PLATE

a shedding plate

4.2.272.2.15

STANDARD PLATE

a standard plate

4.2.272.2.16

STIFFENER

a stiffener in general

4.2.272.2.17

TRANSVERSAL STIFFENER

a transversal stiffener

4.2.272.2.18

USER DEFINED)

user defined functionality

4.2.272.2.19

WATERTIGHT CLIP

a wathertight clip

4.2.272.2.20**WEB**

a web

4.2.273 Structural_part_joint

A `Structural_part_joint` is a type of `Item_relationship` (see 4.2.133) that represents a logical bridge between structural parts. The `Structural_part_joint` may decompose into lower level part joints.

The data associated with a `Structural_part_joint` are the following:

- `joint_type`;
- `realization`.

4.2.273.1 joint_type

The `joint_type` denotes the type of joint represented by a structural part connectivity.

The value of the `joint_type` shall be one of the following:

- `fixed_joint`;
- `movable_joint`.

NOTE See 4.2.273.1.1 - 4.2.273.1.2 for the definition of each allowable value for `joint_type`.

4.2.273.1.1**fixed joint**

a type of structural part joint

NOTE A fixed joint manifests itself as either a mechanical joint or a welded joint. The fixed joint results in a connection between structural parts that is rigid and allows no movement of the parts joined relative to one another.

4.2.273.1.2**movable joint**

a type a type of structural part joint where the parts to be joined are not constrained to fixed positions, but are allowed to move relative to one another, such as in the case of a hinge

NOTE This type of joint is usually accomplished by placing a structural opening in both parts and holding them together with some sort of fastener set.

4.2.273.2 realization

The realization specifies that the `Structural_part_joint` shall be realized by `Structural_part_connection_implementations`

4.2.274 Structural_part_joint_design_definition

A `Structural_part_joint_design_definition` is a type of `Design_definition` (see 4.2.63) that specifies the the design aspects of a `Structural_part_joint` (see 4.2.273), such as its tightness, and its `joint_feature`.

The data associated with a `Structural_part_joint_design_definition` are the following:

- `defined_for`;
- `joint_feature`;
- `tightness`.

4.2.274.1 defined_for

The `defined_for` specifies the redeclaring the `defined_for` attribute in order to allow the `Structural_part_joint_design_definition` to point to a `Structural_part_joint` only. See 4.3.175 for the application assertion.

4.2.274.2 joint_feature

The `joint_feature` specifies the identifier used to denote whether a structural part joint represents a face-face, a face-edge or a edge-edge joint. This identifier is based on the topological decomposition of a `Structural_part_joint`.

The value of the `joint_feature` shall be one of the following:

- `edge edge`.
- `face edge`;
- `face face`;

NOTE See 4.2.274.2.1 - 4.2.274.2.3 for the definition of each allowable value for `joint_feature`.

4.2.274.2.1

edge edge

the edge of one part is joined to the edge of the other

EXAMPLE 1 the side of one plate part joined with the side of another

4.2.274.2.2**face edge**

the face of one part is joined to the edge of the other

EXAMPLE 1 the connection of a bracket to the web of a structural shape

4.2.274.2.3**face face**

the face of one part is joined to the face of the other

EXAMPLE 1 in the case of that two plates lap over each other

4.2.274.3 tightness

The tightness specifies the actual degree of tightness achieved by the structural part joint. The supported tightness levels are air tight, fume tight, weather tight, water tight and oil tight. The actual joint tightness may be greater than that required by the connection tightness.

The value of the tightness shall be one of the following:

- air tight;
- fume tight;
- non tight;
- oil tight;
- undefined;
- water tight;
- weather tight.

NOTE See 4.2.274.3.1 - 4.2.274.3.7 for the definition of each allowable value for tightness.

4.2.274.3.1**air tight**

impermeable to air

4.2.274.3.2**fume tight**

impermeable to fume

4.2.274.3.3**non tight**

not tight

4.2.274.3.4

oil tight

impermeable to oil

4.2.274.3.5

undefined

the tightness is not defined

4.2.274.3.6

water tight

impermeable to water except when under sufficient pressure to produce structural discontinuity

4.2.274.3.7

weather tight

tight under open air condition

4.2.275 Structural_part_manufacturing_definition

A `Structural_part_manufacturing_definition` is a type of `Manufacturing_definition` (see 4.2.154). A `Structural_part_manufacturing_definition` is either a `Plate_manufacturing_definition` (see 4.2.189), or a `Profile_manufacturing_definition` (see 4.2.210). A `Structural_part_manufacturing_definition` specifies the production lifecycle stage view of a structural part that represents characteristics and attributes directly related to the manufacturing of the part. An earlier detailed design part stage view may or may not exist for the part. At this stage, structural part features are identified that affect aspects of the part that have to do with processes that will be used to cut and form the part. Some examples of the kinds of features identified during this stage are edge grinds, layout marks, and part stock. Information relating the structural part to the raw material stock it will be manufactured from is also added during this stage. A 2D flat pattern representation may be defined for the part, enabling it to be nested and cut from flat raw material stock.

The data associated with a `Structural_part_manufacturing_definition` are the following:

- `annotation`;
- `defined_for`;
- `inner_contours`;
- `layout_marks`.

4.2.275.1 annotation

The annotation is the text or symbols that will be marked onto the part during manufacturing to provide part identification or indications of correct part orientation within the ship. See 4.3.176

for the application assertion.

4.2.275.2 defined_for

The `defined_for` specifies the re-declaration of the `defined_for` attribute to be allowed to point to `Structural_parts` only. See 4.3.177 for the application assertion.

4.2.275.3 inner_contour

The `inner_contour` specifies items in the representation that represent holecuts in interior_features that will be burned from the part during manufacturing. See 4.3.176 for the application assertion.

4.2.275.4 layout_marks

The `layout_marks` specifies the items in the representation that will be marked on the part during manufacturing to provide part indications of reference locations or attachments to other structural parts. See 4.3.176 for the application assertion.

4.2.276 Structural_part_penetration_relationship

A `Structural_part_penetration_relationship` is a type of `Item_relationship` (see 4.2.133) that specifies a `Structural_cutout` (see 4.2.264) that results from the fact that a `Part` or `System` penetrates a `Structural_part` (see 4.2.269). The `Structural_part_penetration_relationship` will normally result in one or several cutouts for the involved elements.

EXAMPLE 1 This relationship may be used to describe that a pipe penetrates a profile or that a profile cuts through the edge of a plate.

The data associated with a `Structural_part_penetration_relationship` are the following:

- `item_1`;
- `penetration_result`.

4.2.276.1 item_1

The `Item_1` shall be a `Structural_part`. This is the penetrated Item of the two Items involved in the `Structural_part_penetration_relationship`. `Item_1` will become the parent of a cutout feature, as this relationship results in a cutout. See 4.3.179 for the application assertion.

4.2.276.2 penetration_result

The `penetration_result` specifies the set of `Structural_cutouts` that this penetration results in. See 4.3.178 for the application assertion.

NOTE Normally the result would be just one cutout. However, it is possible that both involved parts may get cutouts. The parent attribute in `Structural_cutout` tells which part gets which cutout. `Item_2` is the Item that is considered the penetrating one - if it at all is possible to distinguish penetrated and penetrating.

4.2.277 `Structural_part_relationship`

A `Structural_part_relationship` is a type of `Part_relationship` (see 4.2.181) that is the AP 218 specific `Part_relationship`. It describes the association of a `Structural_part` (see 4.2.269) with another "structural item". The "Structural item" can be one of

- a) `Structural_system`;
- b) `Structural_part`;
- c) `Structural_feature`.

The data associated with a `Structural_part_relationship` are the following:

— `item_1`.

The `item_1` specifies the `Structural_part_relationship` shall relate a `Structural_part` with something else. See 4.3.180 for the application assertion.

4.2.278 `Structural_part_symmetry_relationship`

A `Structural_part_symmetry_relationship` is a type of `Structural_part_relationship` (see 4.2.277) that specifies that two `Structural_parts` (see 4.2.269) are symmetric with respect to a given plane.

NOTE This relationship is only an informal one for those applications that can use this information. No implicit information is intended to be carried especially this is no short-cut for the definition of one of the related `Structural_parts` using the other one.

The data associated with a `Structural_part_symmetry_relationship` are the following:

- `item_2`;
- `mirroring_plane`.

4.2.278.1 `item_2`

The `item_2` specifies that the `Structural_part_symmetry_relationship` can only be between two `Structural_parts`. See 4.3.182 for the application assertion.

4.2.278.2 mirroring_plane

The mirroring_plane specifies the plane the symmetry is with respect to. See 4.3.181 for the application assertion.

4.2.279 Structural_system

A Structural_system is a type of System (see 4.2.289) that is the AP 218 specific System. It provides information and capabilities common to all types of Structural_systems. The Items carried by a Structural_system are restricted to be of type Structural_system or Structural_part (see 4.2.269). The same holds for the relationships.

Figure 34 is a part of the Structural_system of a ship.

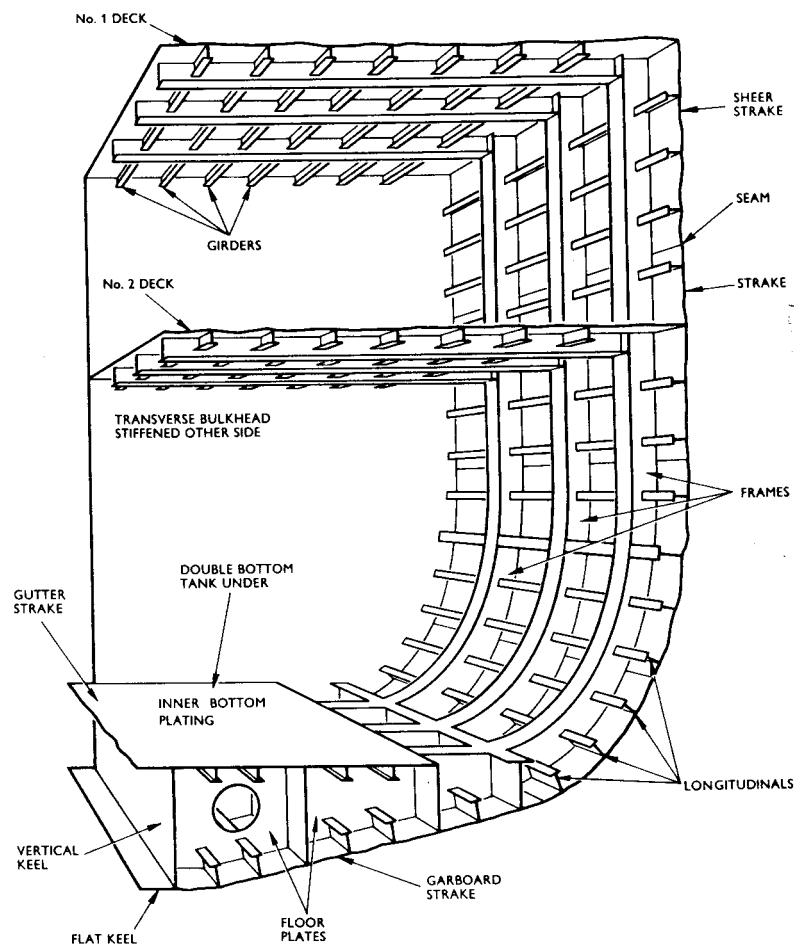


FIG. 3-14 Cross-section of a frigate

Figure 34 – Structural_system

A Structural_system is of a specific class, as specified by the class attribute:

SECTION an aggregate of other (sub)sections, blocks and/or Structural_parts; has both a strength and a tightness function (see Figure 35 and Figure 36).

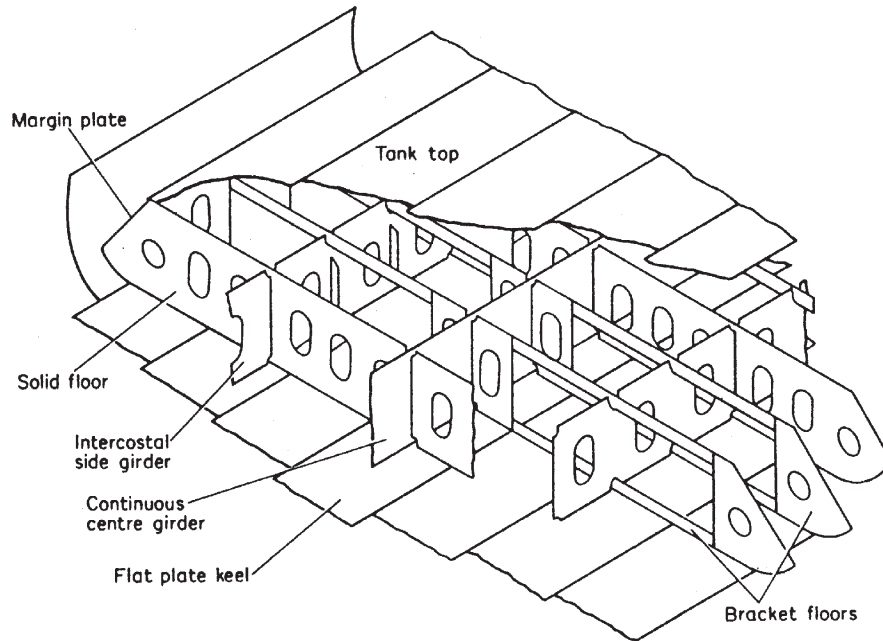


Figure 35 – A doublebottom as section

PANEL_SYSTEM special section that consists of one to many plates reinforced by profiles running in the same main direction; the plates lie in the same (smooth mathematical) moulded surface (plane or non-plane);

CORRUGATED_STRUCTURE special section that consists of only Corrugated_parts and Plates;

BLOCK also referred to as constructional group; an aggregate of other (sub)blocks and/or Structural_parts; has either a strength or a tightness function PLA.

The data associated with a Structural_system are the following:

- class;
- user_def.class.

4.2.279.1 class

The class specifies the structure class allows for a subdivision into different levels of structural systems.

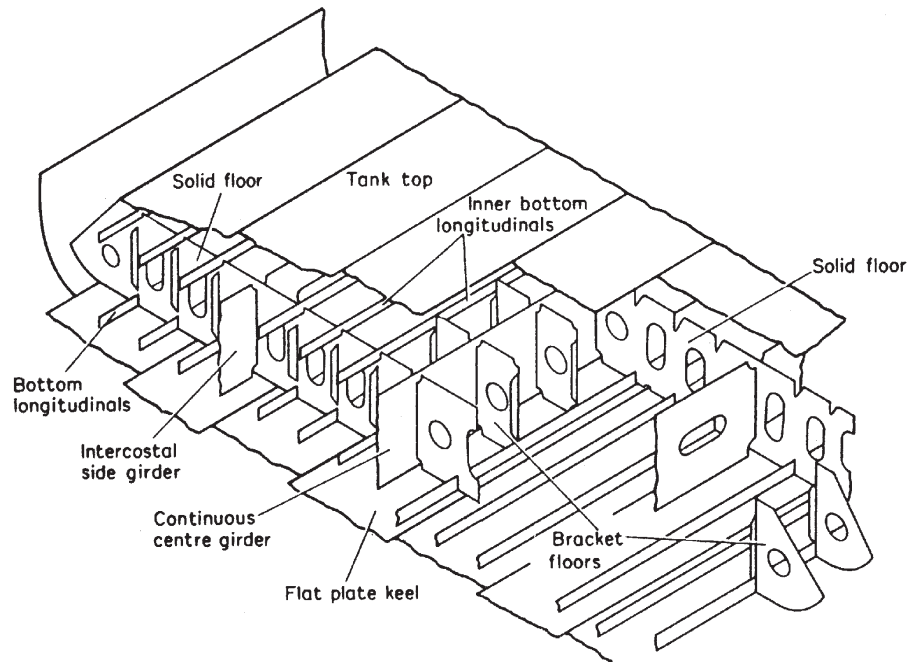


Figure 36 – A doublebottom

4.2.279.2 user_def_class

The `user_def_class` specifies the user defined class of the structural system to be used in case `class = USER_DEFINED`. The `user_def_class` need not to be specified for a particular `Structural_system`.

4.2.280 Structural_system_adjacency_relationship

A `Structural_system_adjacency_relationship` is a type of `Structural_system_relationship` (see 4.2.284) that defines a boundary of a `Structural_system` (see 4.2.279) by specifying a boundary of another `Structural_system` that is adjacent to `SELF` making the related `Structural_systems` sharing the referred boundary and an orthogonal displacement off that boundary.

The data associated with a `Structural_system_adjacency_relationship` are the following:

- `boundary_index`;
- `displacement`;
- `item_2`.

4.2.280.1 boundary_index

The `boundary_index` specifies the index of the referred boundary in the boundary list of the relevant `Structural_system_design_definition` subtype associated with `item_2`

4.2.280.2 displacement

The displacement specifies the displacement specifying the offset orthogonal off the boundary referred to by `boundary_index`

NOTE This is to cover the fact that the two panels are just virtually adjacent but with a gap in between. This allows two `Structural_systems` not only to share boundaries (`displacement = 0`) but to have parallel boundaries where a change to the related boundary automatically changes the dependent one.

4.2.280.3 item_2

The `item_2` specifies the bound Item shall be of type `Structural_system`. See 4.3.183 for the application assertion.

4.2.281 Structural_system_design_definition

The `Structural_system_design_definition` is a type of `System_design_definition` (see 4.2.290) that describes the definition concept of any `Structural_system` (see 4.2.279) from the view of design.

The data associated with a `Structural_system_design_definition` are the following:

- `defined_for`;
- `representations`;
- `tightness`;
- `user_defined_tightness`.

4.2.281.1 defined_for

The `defined_for` specifies the `Structural_system_design_definitions` shall only be declared for `Structural_systems`. See 4.3.185 for the application assertion.

4.2.281.2 representations

The `representations` specifies the `Structural_system_design_definition` shall be represented by `Shape_representations`. See 4.3.184 for the application assertion.

4.2.281.3 tightness

The tightness specifies the defines tightness requirements for a Structural_system.

The value of the tightness shall be one of the following:

- AIR TIGHT;
- FUME TIGHT;
- GAS TIGHT;
- NON TIGHT;
- OIL TIGHT;
- UNSPECIFIED;
- USER DEFINED.
- WATER TIGHT;
- WEATHER TIGHT;

4.2.282 Structural_system_functional_definition

A Structural_system_functional_definition is a type of Functional_definition (see 4.2.112) that defines the functional role of a Structural_system (see 4.2.279); the role may be a pre-defined one or may be user-defined.

The data associated with a Structural_system_functional_definition are the following:

- defined_for;
- function.

4.2.282.1 defined_for

The defined_for specifies the redefinition of the defined_for attribute, a Structural_system_functional_definition is only valid for one or multiple Structural_systems. See 4.3.186 for the application assertion.

4.2.282.2 function

The function specifies the name of a role that a Structural_system may play in a ship.

The value of the function shall be one of the following:

- SHIP STRUCTURE,
- DECK,
- STRENGTH DECK,
- ACCOMODATION DECK,
- MAIN DECK,
- WEATHER DECK,
- NAVIGATION DECK,
- PLATFORM DECK,
- DECK IN SUPERSTRUCTURE,
- BOTTOM,
- INNER BOTTOM,
- DOUBLE BOTTOM,
- OUTER SHELL,
- INNER SHELL,
- DOUBLE SHELL,
- BULKHEAD,
- LONGITUDINAL BULKHEAD,
- TRANSVERSAL BULKHEAD,
- STRENGTH BULKHEAD,
- WASH BULKHEAD,
- COLLISION BULKHEAD,
- HOLD BULKHEAD,
- TANK BULKHEAD,
- WING BULKHEAD,

- SUPERSTRUCTURE FRONT BULKHEAD,
- SUPERSTRUCTURE SIDE BULKHEAD,
- SUPERSTRUCTURE AFT BULKHEAD,
- STOOL,
- WALL,
- PLATING,
- TRANSOM,
- DECK HOUSE,
- SUPERSTRUCTURE,
- MID SHIP,
- FORE SHIP,
- AFT SHIP,
- ENGINE AREA,
- CARGO AREA,
- ACCOMODATION AREA,
- MACHINERY CASING,
- SHEER STRAKE,
- TANK BOTTOM,
- TANK SIDE,
- TANK TOP,
- HATCH COVER,
- KEEL,
- DUCT KEEL,
- HATCHWAY COAMING,

- HATCHWAY SIDECOAMING,
- HATCHWAY ENDCOAMING,
- UPPER BOOM,
- LOWER BOOM,
- FRAME,
- WEB FRAME,
- VERTICAL WEB FRAME,
- TRANSVERSE WEB FRAME,
- FLOOR,
- TRANSVERSE FLOOR,
- CROSS TIE,
- STRINGER,
- GIRDER,
- LONGITUDINAL GIRDER,
- DECK BEAM,
- ENGINE FOUNDATION,
- BRACKET,
- USER DEFINED.

NOTE See 4.2.282.2.1 - 4.2.282.2.66 for the definition of each allowable value for function.

4.2.282.2.1

SHIP STRUCTURE

the whole ship structure

4.2.282.2.2

DECK

a deck in general

Figure 37 is an example for a DECK.

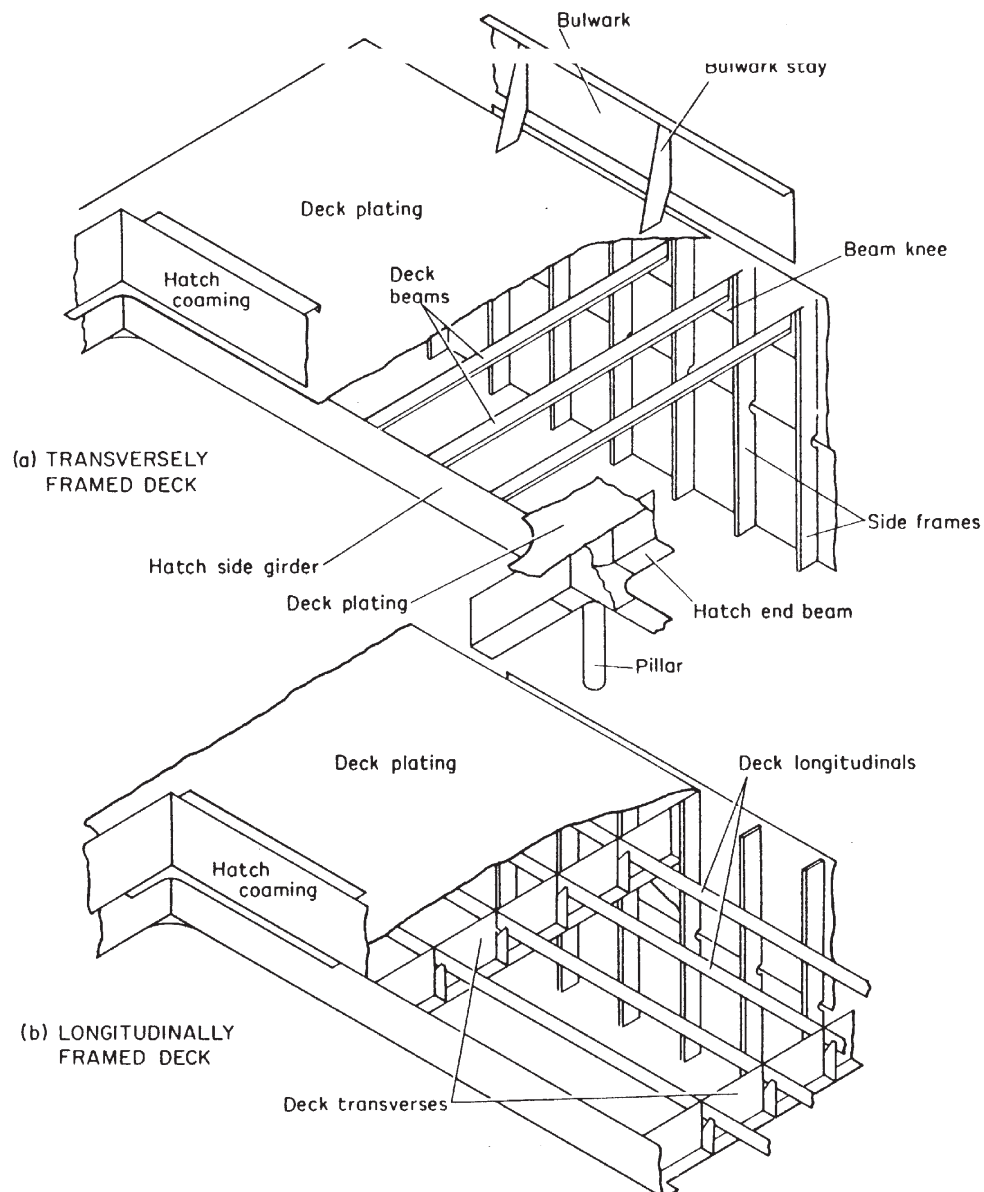


Figure 37 – A DECK

4.2.282.2.3

STRENGTH DECK

a strength deck

4.2.282.2.4

ACCOMODATION DECK

an accomodation deck

4.2.282.2.5

MAIN DECK

a main deck

4.2.282.2.6

WEATHER DECK

a wether deck

4.2.282.2.7

NAVIGATION DECK

a navigation deck

4.2.282.2.8

PLATFORM DECK

a platform deck

4.2.282.2.9

DECK IN SUPERSTRUCTURE

a deck in a superstructure

4.2.282.2.10

BOTTOM

a bottom shell in general

Figure 38 is an example for a single bottom.

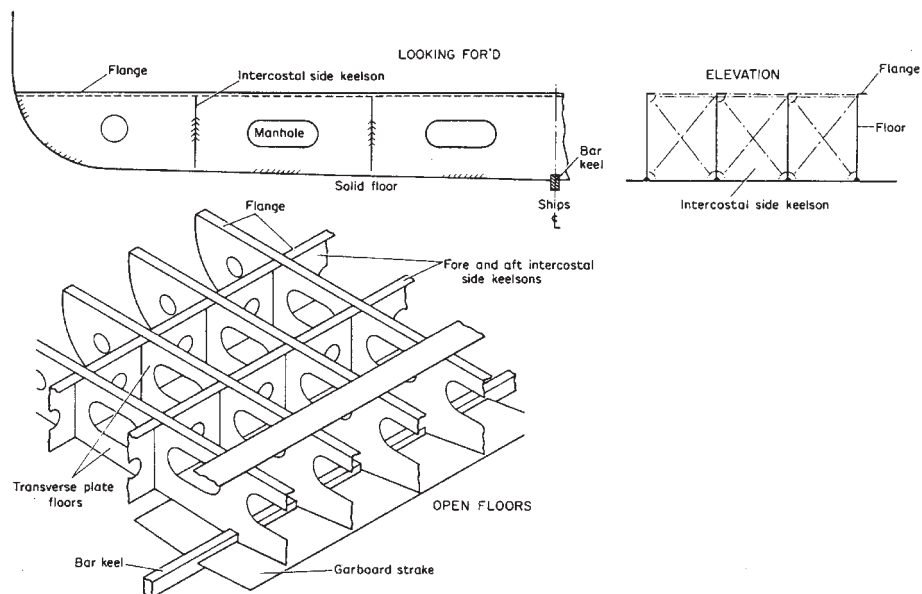


Figure 38 – A single bottom

4.2.282.2.11**INNER BOTTOM**

an inner bottom shell

4.2.282.2.12**DOUBLE BOTTOM**

a double bottom

Figure 39 is an example for a double bottom.

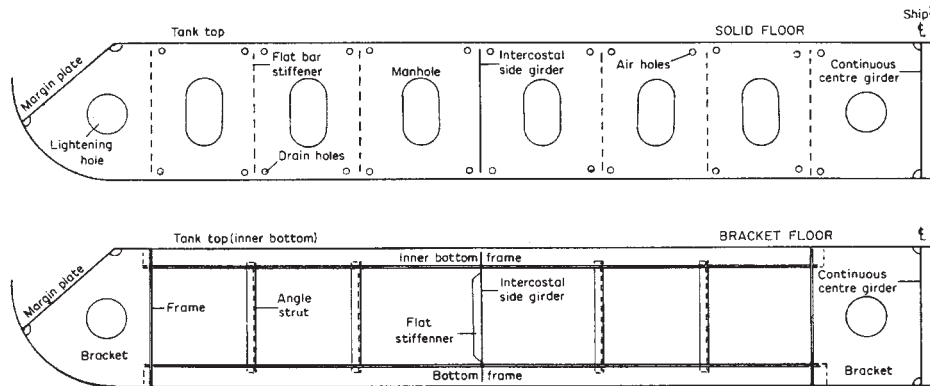


Figure 39 – A double bottom

4.2.282.2.13**OUTER SHELL**

a outer shell

4.2.282.2.14**INNER SHELL**

a inner (double side) shell

4.2.282.2.15**DOUBLE SHELL**

a double shell

Figure 40 is an example for a SHELL.

4.2.282.2.16**BULKHEAD**

a bulkhead in general

4.2.282.2.17**LONGITUDINAL BULKHEAD**

a longitudinal bulkhead

4.2.282.2.18

TRANSVERSAL BULKHEAD

a transversal bulkhead

4.2.282.2.19

STRENGTH BULKHEAD

a strength bulkhead

4.2.282.2.20

WASH BULKHEAD

a wash bulkhead

4.2.282.2.21

COLLISION BULKHEAD

a collision bulkhead

4.2.282.2.22

HOLD BULKHEAD

a hold bulkhead

4.2.282.2.23

TANK BULKHEAD

a tank bulkhead

4.2.282.2.24

WING BULKHEAD

a wing bulkhead

4.2.282.2.25

SUPERSTRUCTURE FRONT BULKHEAD

a superstructure front bulkhead

4.2.282.2.26

SUPERSTRUCTURE SIDE BULKHEAD

a superstructure side bulkhead

4.2.282.2.27

SUPERSTRUCTURE AFT BULKHEAD

a superstructure aft bulkhead

4.2.282.2.28

STOOL

a stool in general

4.2.282.2.29

WALL

a wall in general

4.2.282.2.30

PLATING

a plating in general

4.2.282.2.31

TRANSOM

a transom

4.2.282.2.32

DECK HOUSE

a deck house

4.2.282.2.33

SUPERSTRUCTURE

a superstructure

4.2.282.2.34

MID SHIP

a mid ship

4.2.282.2.35

FORE SHIP

a fore ship

Figure 41 is an example for a FORE SHIP in 2D.

Figure 42 is an example for a FORE SHIP in 3D.

4.2.282.2.36

AFT SHIP

a aft ship

4.2.282.2.37

ENGINE AREA

a engine area

Figure 43 is an example for a bow.

4.2.282.2.38

CARGO AREA

a cargo area

4.2.282.2.39

ACCOMODATION AREA

a accomodation area

4.2.282.2.40

MACHINERY CASING

a machinery casing

4.2.282.2.41

SHEER STRAKE

a sheer strake

4.2.282.2.42

TANK BOTTOM

a tank bottom

Figure 44 is an example for a TANK.

4.2.282.2.43

TANK SIDE

a tank side

4.2.282.2.44

TANK TOP

a tank top

4.2.282.2.45

HATCH COVER

a hatch cover

4.2.282.2.46

KEEL

a keel in general

Figure 45 is an example for a KEEL.

4.2.282.2.47

DUCT KEEL

a duct keel

4.2.282.2.48

HATCHWAY COAMING

a hatchway coaming in general

4.2.282.2.49

HATCHWAY SIDECOAMING

a hatchway sidecoaming

4.2.282.2.50

HATCHWAY ENDCOAMING

a hatchway endcoaming

4.2.282.2.51

UPPER BOOM

a upper boom

4.2.282.2.52

LOWER BOOM

a lower boom

4.2.282.2.53

FRAME

a frame in general

4.2.282.2.54

WEB FRAME

a web frame in general

Figure 46 is an example for a WEB FRAME.

4.2.282.2.55

VERTICAL WEB FRAME

a vertical web frame

4.2.282.2.56

TRANSVERSE WEB FRAME

a transverse web frame

4.2.282.2.57

FLOOR

a floor in general

4.2.282.2.58

TRANSVERSE FLOOR

a transverse floor

4.2.282.2.59

CROSS TIE

a cross tie

4.2.282.2.60

STRINGER

a stringer

4.2.282.2.61

GIRDER

a girder in general

4.2.282.2.62

LONGITUDINAL GIRDER

a longitudinal girder

Figure 47 is an example for a LONGITUDINAL GIRDER.

4.2.282.2.63

DECK BEAM

a deck beam

4.2.282.2.64

ENGINE FOUNDATION

a engine foundation

4.2.282.2.65

BRACKET

a bracket

4.2.282.2.66

USER DEFINED

user defined

4.2.283 Structural_system_penetration_relationship

A `Structural_system_penetration_relationship` is a type of `Item_relationship` (see 4.2.133) that specifies a `Structural_cutout` (see 4.2.264) that results from the fact that a `Part` (see 4.2.179) or `System` (see 4.2.289) penetrates a `Structural_system` (see 4.2.279). The `Structural_system_penetration_relationship` will normally result in one or several cutouts for the involved elements.

EXAMPLE 1 This relationship may be used to describe that a pipe penetrates a panel or that a profile cuts through the edge of a bulkhead. Also a plate group penetrating another plate group is a valid example.

The data associated with a `Structural_system_penetration_relationship` are the following:

- `item_1`;
- `penetration_result`.

4.2.283.1 item_1

The Item_1 shall be a Structural_system. This is the penetrated Item of the two Items involved in the Structural_system_penetration_relationship. Item_1 will become the parent of a cutout feature, as this relationship results in a cutout. See 4.3.188 for the application assertion.

4.2.283.2 penetration_result

The penetration_result specifies the set of Structural_cutouts that this penetration results in.

NOTE Normally the result would be just one cutout. However, it is possible that both involved parts may get cutouts. The parent attribute in Structural_cutout tells which part gets which cutout. Item_2 is the Item that is considered the penetrating one - if it at all is possible to distinguish penetrated and penetrating.

See 4.3.187 for the application assertion.

4.2.284 Structural_system_relationship

A Structural_system_relationship is a type of System_relationship (see 4.2.291). It describes the association of a Structural_system with another "structural item". "Structural item" can be one of

- a) Structural_system (see 4.2.279)
- b) Structural_part (see 4.2.269)
- c) Structural_feature (see 4.2.266).

The data associated with a Structural_system_relationship are the following:

— item_1.

The item_1 specifies the Structural_system_relationship shall relate a Structural_system with something else. See 4.3.189 for the application assertion.

4.2.285 Structural_system_symmetry_relationship

A Structural_system_symmetry_relationship is a type of Structural_system_relationship (see 4.2.284) that specifies that two Structural_systems (see 4.2.279) are symmetric with respect to a given plane.

NOTE This relationship is only an informal one for those applications that can use this information. No implicit information is intended to be carried especially this is no short-cut for the definition of one of the related Structural_systems using the other one.

The data associated with a `Structural_system_symmetry_relationship` are the following:

- `item_2`;
- `mirroring_plane`.

4.2.285.1 `item_2`

The `item_2` specifies the `Structural_system` to which the referencing one is said to be symmetric to. See 4.3.191 for the application assertion.

4.2.285.2 `mirroring_plane`

The `mirroring_plane` specifies the plane the symmetry is with respect to. See 4.3.190 for the application assertion.

4.2.286 `Structural_weld_shrinkage_allowance_feature`

A `Structural_weld_shrinkage_allowance_feature` is a type of `Structural_manufacturing_feature` (see 4.2.268) that is a type of part feature that represents the amount by which the design definition of the part is to be scaled in the directions of the three primary axes of the ship to account for later shrinkage due to welding of the part into an assembly. Application of the `Structural_weld_shrinkage_allowance_feature` to the design definition results in a manufacturing definition representation whose geometry is scaled up in one or more of the primary ship axes directions.

4.2.287 `Structural_weld_shrinkage_allowance_feature_design_definition`

A `Structural_weld_shrinkage_allowance_feature_design_definition` is a type of `Feature_design_definition` (see 4.2.96) that provides the values by which the manufactured part is to be scaled to account for later shrinkage due to welding of the part into an assembly. Values are provided for the longitudinal, transverse, and vertical dimensions relative to the ship's global coordinate system. A `Structural_weld_shrinkage_allowance_feature` will scale the edge geometry and marking features of the part in either or both directions, resulting in a new set of edges that represent the "extended" part's boundary for burning during manufacturing of the part and new locations for marking lines and the centers of holecuts.

EXAMPLE 1 No value means that the part size need not be increased to allow for weld shrinkage. A value of 0.0003 means that the part and all of its features shall be scaled by .3 mm per meter of part length in the direction for which an allowance is specified.

The data associated with a `Structural_weld_shrinkage_allowance_feature_design_definition` are the following:

- `defined_for`.

The `defined_for` specifies the `Structural_weld_shrinkage_allowance_feature` that the `Structural_weld_shrinkage_allowance_feature_design_definition` is defined for. See 4.3.192 for the application assertion.

4.2.288 Supplier_BSU

A `Supplier_BSU` (supplier basic semantical unit) is a type of `BSU` (see 4.2.19) that identifies the supplier of a parts library.

The data associated with a `Supplier_BSU` are the following:

— `code`.

The code specifies the designation of the identification of a parts library supplier. The supplier code shall be unique.

4.2.289 System

A `System` is a type of both `Item` (see 4.2.132) and `Item_structure` (see 4.2.134) that is a functional unit on board a ship. A `System` is a collection of `Parts` and `Systems`. `System` itself is independent of any discipline. Sub-types may, however, be created to represent for example structural systems, piping systems, or propulsion systems. `Systems` have tree structures and shall not be cyclic. This means that a `System` shall not be referenced by itself directly or indirectly, that is via the relationships that it consists of.

4.2.290 System_design_definition

A `System_design_definition` is a type of `Design_definition` (see 4.2.63) that describes aspects of one or many `Systems` (see 4.2.289). Such `Definitions` shall collect properties that are specific to the life-cycle phase of design.

The data associated with a `System_design_definition` are the following:

— `defined_for`.

The `defined_for` specifies the `Systems` that a `System_design_definition` shall be valid for. See 4.3.193 for the application assertion.

4.2.291 System_relationship

A `System_relationship` is a type of `Item_relationship` (see 4.2.133) that describes the association of a `System` (see 4.2.289) and either another `System` or a `Part` (see 4.2.179).

NOTE Subtypes of this entity may for example be used to specify the weld seams necessary to connect an `Item` to a `System`.

The data associated with a `System_relationship` are the following:

- `item_1`.

4.2.291.1 item_1

The `item_1` specifies the System that is related to a Part or another System. See 4.3.194 for the application assertion.

4.2.292 T_bar_cross_section

A `T_bar_cross_section` is a type of `Flanged_profile_cross_section` (see 4.2.100) that consists of one flange and one web. The web is centered on the flange.

NOTE ISO 657-21 specifies dimensions for some standard T-bar sections.

The data associated with a `T_bar_cross_section` are the following (see Figure 48):

- `depth`;
- `flange_thk`;
- `k`;
- `radius`;
- `web_thk`;
- `width`.

4.2.292.1 depth

The `width` specifies the depth of the section.

4.2.292.2 flange_thk

The `flange_thk` specifies the flange thickness.

4.2.292.3 k

The `k` specifies the distance from outside flange to fillet on web.

4.2.292.4 radius

The `width` specifies the radius between the web and the flange.

4.2.292.5 web_thk

The width specifies the web thickness of the T_bar_cross_section.

4.2.292.6 width

The width specifies the the flange width.

4.2.293 Thermodynamic_temperature_measure_with_unit

A Thermodynamic_temperature_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a thermodynamic_temperatur as defined in ISO 31 (clause 2).

4.2.294 Thermodynamic_temperature_unit

A Thermodynamic_temperature_unit is a type of Named_unit (see 4.2.164) that is the unit in which the degree of heat of a body is measured.

4.2.295 Time_measure_with_unit

A Time_measure_with_unit is a type of Extended_measure_with_unit (see 4.2.91) where the physical quantity is a time as defined in ISO 31 (clause 2).

4.2.296 Time_unit

A Time_unit is a type of Named_unit (see 4.2.164) that is the unit in which the duration of periods is measured.

4.2.297 Transversal_position

A Transversal_position is a type of Spacing_position (see 4.2.246) that is a spacing position located on the global y axis.

4.2.298 Transversal_table

A Transversal_table is a type of Spacing_table (see 4.2.248) that is a spacing table whose positions lie on the transverse axis of the global co-ordinate system, which is the global y axis.

The data associated with a Transversal_table are the following:

— spacing_table_representations.

The spacing_table_representations specifies the transversal positions which make up the transversal table. See 4.3.195 for the application assertion.

4.2.299 Triangular_cutout_design_definition

A `Triangular_cutout` is a type of `Interior_cutout_design_definition` (see 4.2.128) that is characterised by the connection of three linear independent points in a plane. The origin is placed at corner point `c1`, and the x-axis is defined by the straight connection from corner point `c1` to `c2`. The y-axis is perpendicular to the x-axis and such that corner point `c3` has a positive y-coordinate.

The data associated with a `Triangular_cutout_design_definition` are the following:

- `c1_radius`;
- `c2_radius`;
- `c3_radius`;
- `c2_x`;
- `c3_x`;
- `c3_y`.

4.2.299.1 c1_radius

The `c1_radius` specifies the radius of the circular arc that rounds corner `c1`.

4.2.299.2 c2_radius

The `c2_radius` specifies the radius of the circular arc that rounds corner `c2`.

4.2.299.3 c3_radius

The `c3_radius` specifies the radius of the circular arc that rounds corner `c3`.

4.2.299.4 c2_x

The `c2_x` specifies the x-coordinate of corner point `c2`.

4.2.299.5 c3_x

The `c3_x` specifies the x-coordinate of corner point `c3`.

4.2.299.6 c3_y

The `c3_y` specifies the y-coordinate of corner point `c3`.

4.2.300 Twist_location

A Twist_location is an explicit location and orientation along the trace line of a profile where an angle of twist is to be specified.

The data associated with a Twist_location are the following:

- location;
- orientation.

4.2.300.1 location

The location specifies the location on a Profile design definition trace line where an angle of twist is to be specified. See 4.3.197 for the application assertion.

4.2.300.2 orientation

The orientation specifies the angle of the cross-section's local v axis, tangent to the web of the profile, at the location, for a profile that twists along its length. See 4.3.196 for the application assertion.

4.2.301 Universal_resource_locator

A Universal_resource_locator is the address of an electronic data source (i.e. an Internet address). This is an alternative to the common mail address as provided by ISO 10303-41.

The data associated with an Universal_resource_locator are the following:

- location;
- machine_address;
- other_protocol_type;
- port;
- protocol.

4.2.301.1 location

The location specifies the path on the target machine where the document is located.

4.2.301.2 machine_address

The machine_address specifies the name of the target machine that provides the service (i.e. the ftp server name or an Internet address).

4.2.301.3 other_protocol_type

The `other_protocol_type` gives the possibility to specify a transmission protocol apart from the ones of `Protocol_type`. The `other_protocol_type` need not to be specified for a particular `Universal_resource_locator`.

4.2.301.4 port

The `port` specifies for some protocols (i.e. `http`), it is possible to use an other port-number then the standard one. If so it can be specified here. The `port` need not to be specified for a particular `Universal_resource_locator`.

4.2.301.5 protocol

The `protocol` specifies the type of the transmission protocol (i.e. `ftp`, `http`, etc.) Because there is a permanent development on this field it is not possible to cover all available protocol types with the enumeration. Because of this a protocol type `USER_DEFINED` is included and the optional `other_protocol_type` attribute is used to hold the protocol type in this case. A `WHERE` rule assures that in case of `protocol = USER_DEFINED` the `other_protocol_type` attribute is in use.

The value of the `protocol` shall be one of the following:

- `FTP`;
- `HTTP`;
- `USER_DEFINED`.

NOTE See 4.2.301.5.1 - 4.2.301.5.3 for the definition of each allowable value for `protocol`.

4.2.301.5.1**FTP file**

transfer protocol

4.2.301.5.2**HTTP**

hyper text transfer protocol

4.2.301.5.3**USER_DEFINED**

user defined transfer protocol the different types of transfer protocols

4.2.302 Version_creation

A `Version_creation` is a type of `Versionable_object_change_event` (see 4.2.308) that is the event leading to a new `Definition`, `Item_structure`, or `Item_relationship`. The base attribute shall

be populated if the subject `Versionable_object` is based upon an existing `Versionable_object`. The base `Versionable_object` need not be the immediately preceding version of the subject `Versionable_object`, but may refer to any previous version in the version history of the same Item, or to any `Versionable_object` of another Item which contributes to the creation of the subject `Versionable_object`.

The data associated with a `Version_creation` are the following:

- `base`;
- `subject`.

4.2.302.1 base

The base specifies the `Versionable_objects` the subject is derived from. See 4.3.198 for the application assertion.

4.2.302.2 subject

The subject specifies the `Versionable_objects` created by the change event. See 4.3.198 for the application assertion.

4.2.303 Version_deletion

The `Version_deletion` is a type of `Versionable_object_change_event` (see 4.2.308) that is the event leading to the deletion of a `Definition`, an `Item_structure`, or an `Item_relationship`.

The data associated with a `Version_deletion` are the following:

- `subject`.

The subject specifies the `Versionable_object` deleted or to be deleted by the change event. See 4.3.199 for the application assertion.

4.2.304 Version_history

A `Version_history` identifies `Versionable_objects` (see 4.2.307) and their `Version_relationships` (see 4.2.306) in terms of their role as predecessors respectively successors with respect to each other. The `Version_history` shall be a directed acyclic graph. Consequently the `Version_history` may contain `Versionable_objects` considered alternatives with respect to each other (a `Versionable_object` having more than one successor), and merged `Versionable_objects` (a `Versionable_object` having more than one predecessor).

The data associated with a `Version_history` are the following:

- `current_version`;

- relationships;
- versions.

4.2.304.1 current_version

The `current_version` specifies the `Versionable_object` which plays the role of the `Current_version` in this `Version_history`. See 4.3.201 for the application assertion.

4.2.304.2 relationships

The `relationships` specifies the `Version_relationships` the `Version_history` consists of. See 4.3.200 for the application assertion.

4.2.304.3 versions

The `versions` specifies the `Versionable_objects` the `Version_history` consists of. See 4.3.201 for the application assertion.

4.2.305 Version_modification

The `Version_modification` is a type of `Versionable_object_change_event` (see 4.2.308) that is the event leading to a change of a `Versionable_object`, i.e., the creation of a new version for an existing thing. The base `Versionable_object` need not be the immediately preceding version of the subject `Versionable_object`, but may refer to any previous version in the version history of the same `Item`.

The data associated with a `Version_modification` are the following:

- `base`;
- `subject`.

4.2.305.1 base

The `base` specifies the `Versionable_objects` the `subject` is derived from. See 4.3.202 for the application assertion.

4.2.305.2 subject

The `subject` specifies the `Versionable_objects` modified or to be modified by the change event. See 4.3.202 for the application assertion.

4.2.306 Version_relationship

A Version_relationship defines the relationship of two Versionable_objects (see 4.2.306) of same type in terms of a Version_history (see 4.2.304).

The data associated with a Version_relationship are the following:

- predecessor;
- reason;
- successor.

4.2.306.1 predecessor

The predecessor specifies the Version the successor is derived from. See 4.3.203 for the application assertion.

4.2.306.2 reason

The reason specifies the reason for a new version, created by a certain person at a certain time.

4.2.306.3 successor

The successor specifies the Version the predecessor is the preceding version of. See 4.3.203 for the application assertion.

4.2.307 Versionable_object

A Versionable_object is either a Definition (see 4.2.57), an Item_relationship (see 4.2.133), or an Item_structure (see 4.2.134).

The data associated with a Versionable_object are the following:

- version_id.

The version_id specifies the version of the instance of itself.

4.2.308 Versionable_object_change_event

A Versionable_object_change_event is a type of Event (see 4.2.86) that is the generalization of the events effectively changing a Definition, Item_structure, or Item_relationship.

4.2.309 Vertical_position

A `Vertical_position` is a type of `Spacing_position` (see 4.2.246) that is a spacing position located on the global z axis.

4.2.310 Vertical_table

A `Vertical_table` is a type of `Spacing_table` (see 4.2.248) that is a spacing table whose positions lie on the vertical axis of the global co-ordinate system, which is the global z axis.

The data associated with a `Vertical_table` are the following:

— `spacing_table_representations`.

The `spacing_table_representations` specifies the vertical positions which make up the vertical table. See 4.3.204 for the application assertion.

4.2.311 Voltage_measure_with_unit

A `Voltage_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is voltage.

4.2.312 Voltage_unit

A `Voltage_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which electric potential is measured.

4.2.313 Volume_measure_with_unit

A `Volume_measure_with_unit` is a type of `Extended_measure_with_unit` (see 4.2.91) where the physical quantity is a volume as defined in ISO 31 (clause 2).

4.2.314 Volume_unit

A `Volume_unit` is a type of `Named_unit` (see 4.2.164) that is the unit in which the solid content of a body is measured.

4.2.315 W_shape_cross_section

A `W_shape_cross_section` is a type of `Flanged_profile_cross_section` (see 4.2.100) commonly known as a Wide flange shape. It is distinguished from other standard beam and column shapes by having non-sloping flanges.

The data associated with a `W_shape_cross_section` are the following:

— `depth`;

- flange_thk;
- k;
- radius;
- web_thk;
- width.

4.2.315.1 depth

The width specifies the depth at the outside of the flanges.

4.2.315.2 flange_thk

The flange_thk specifies the flange thickness of the W_shape_cross_section.

4.2.315.3 k

The k specifies the distance from outside flange to fillet on web.

4.2.315.4 radius

The width specifies the radius between the web and the flange.

4.2.315.5 web_thk

The width specifies the web thickness of the W_shape_cross_section.

4.2.315.6 width

The width specifies the width of the flanges.

4.2.316 Waterline_table

A Waterline_table is a type of Vertical_table (see 4.2.310) whose positions are a reference for the location of waterlines and are located on the global z axis.

The data associated with a Waterline_table are the following:

- spacing_table_representations.

The spacing_table_representations specifies the vertical positions of the waterlines.

4.2.317 Weight_and_centre_of_gravity

A `Weight_and_centre_of_gravity` specifies the weight and possibly the centre of gravity of a ship part. The moment components will be derived if the moment origin exists.

The data associated with a `Weight_and_centre_of_gravity` are the following:

- `centre_of_gravity`;
- `mass`;
- `moment`.

4.2.317.1 centre_of_gravity

The `centre_of_gravity` specifies the centre of gravity of a ship part. See 4.3.205 for the application assertion.

4.2.317.2 mass

The `mass` specifies the weight of a ship part.

4.2.317.3 moment

The `moment` specifies the moment based on `centre_of_gravity` and `mass`. See 4.3.206 for the application assertion.

4.2.318 Weld

A `Weld` is a type of `Item` (see 4.2.132) that inherits all attributes of an `Item`: the name, the descriptions and documentations.

Figure 49 is an example of welds.

4.2.319 Weld_design_definition

A `Weld_design_definition` is a type of `Design_definition` (see 4.2.63) that specifies the weld definition from the design lifecycle stage view of a weld that represents characteristics and attributes directly related to the weld design of the parts. At this stage the welding features are identified that affect aspects of the parts that have to do with processes that will be used to cut and form the parts for the purpose of welding preparations.

The data associated with a `Weld_design_definition` are the following:

- `connection_angle`;

- `defined_for`;
- `end_suspension`;
- `inclination_angle`;
- `rotation_angle`;
- `shape_of_weld_surface`;
- `shrinkage`;
- `start_suspension`;
- `weld_cross_section_area`;
- `weld_geometry`;
- `weld_size`.

4.2.319.1 connection_angle

The `connection_angle` specifies the angle between two welded parts at the connecting point.

4.2.319.2 defined_for

The `defined_for` specifies the redeclaring the `defined_for` attribute in order to allow the `Weld-manufacturing_definition` to point to a `Weld` only. See 4.3.209 for the application assertion.

4.2.319.3 end_suspension

The `end_suspension` specifies the suspension of welding at the end point of the `Weld`.

4.2.319.4 inclination_angle

The `inclination_angle` specifies the angle between the horizontal plane and the root line of welding. The value of the `inclination_angle` shall be variable from 0 to 90 degrees.

4.2.319.5 rotation_angle

The `rotation_angle` specifies the angle between the vertical plane including the root axis, and the plane including both the root axis and the line equally distant from the groove faces, or the line dividing equally the angle between two plates in case of fillet welding. The value of the `rotation_angle` shall be 180 degree or under.

4.2.319.6 shape_of_weld_surface

The shape_of_weld_surface specifies the resulted top surface shape of a Weld.

The value of the shape_of_weld_surface shall be one of the following:

- flush;
- convex;
- concave;
- none.

NOTE See 4.2.319.6.1 - 4.2.319.6.4 for the definition of each allowable value for shape_of_weld_surface.

4.2.319.6.1

flush

forming a continuous plane top surface at a weld

4.2.319.6.2

convex

forming a convex top surface at a weld

4.2.319.6.3

concave

forming a concave top surface at a weld

4.2.319.6.4

none

none top surface shape type

4.2.319.7 shrinkage

The shrinkage specifies the proposed shrinkage of the parts at the Weld.

4.2.319.8 start_suspension

The start_suspension specifies the suspension of welding at the start point of the Weld.

4.2.319.9 weld_cross_section_area

The weld_cross_section_area specifies the area measure of the cross-section of the weld detail to be filled with weld filler material. This area can be used to estimate the amount of weld material required. The weld_cross_section_area need not to be specified for a particular Weld_design_definition.

4.2.319.10 weld_geometry

The weld_geometry specifies the geometry of the connected contour of welded parts, which references ISO 10303/part 501. See 4.3.208 for the application assertion.

4.2.319.11 weld_size

The weld_size specifies the dimensions used in design to designate the result size of a Weld, e.g. leg length of fillet weld, and width as well as reinforcement height of butt weld. Cases of equal legs and unequal legs, concave and convex surface forms, etc. are considered. The weld_size need not to be specified for a particular Weld_design_definition.

4.2.320 Weld_filler_material

A Weld_filler_material is a type of Ship_material_property (see 4.2.242) that represents the chemical composition and make-up of metal being used to fill the joint opening. The final strength of welded joint depends on the type of weld fillet material used.

The data associated with a Weld_filler_material are the following:

- chemical_composition;
- chemical_composition_id;
- defined_for;
- electrode_id;
- material_strength_id;
- notch_impact_work_id.

4.2.320.1 chemical_composition

The chemical_composition specifies the chemical composition of the wire electrode, it includes the percentage value of the identified alloy components. See 4.3.210 for the application assertion.

4.2.320.2 chemical_composition_id

The chemical_composition_id specifies the identifier of the chemical composition of the wire electrode.

4.2.320.3 defined_for

The defined_for specifies the redeclaring the defined_for attribute in order to allow the Weld_filler_material, as a type of definition, to point to Weld only. See 4.3.211 for the application assertion.

4.2.320.4 electrode_id

The `electrode_id` specifies the identifier for the used electrode wire which provides filler material to the welded joint. For example, G0, G3Si1, etc.

4.2.320.5 material_strength_id

The `material_strength_id` specifies the strength identifier for the weld filler material. It specifies the minimum extensibility, the tensile strength and the minimum fracture-stretch of the weld filler material.

EXAMPLE 1 A filler material has a `material_strength_id` "35", it means that the material has minimum extensibility 355 N/mm², tensile strength 440 to 570 N/mm² and minimum fracture-stretch 22%.

4.2.320.6 notch_impact_work_id

The `notch_impact_work_id` specifies the identifier for the notch impact work of the filler material. It is defined by a temperature at which the notch impact work reaches a value of 47J. For example, a filler material has a `notch_impact_work_id` "A", it means that the temperature for the minimum notch_impact_work 47J is +20 degree C.

4.2.321 Weld_manufacturing_definition

A `Weld_manufacturing_definition` is a type of `Manufacturing_definition` (see 4.2.154) that specifies the weld definition from the production lifecycle stage view of a weld that represents characteristics and attributes directly related to the weld manufacturing of the parts. At this stage the welding methods, environment, procedures and etc. are specified that affect aspects of the weld manufacturing processes and the qualification of a weld.

The data associated with a `Weld_manufacturing_definition` are the following:

- `defined_for`;
- `degree_of_automatizations`;
- `environment`;
- `number_of_weld_passes`;
- `penetration`;
- `position`;
- `process`;
- `sequences`;

- tack_weld_used;
- torch_vector;
- weld_test;
- welding_deposition_sequences;
- welding_procedures.

4.2.321.1 defined_for

The defined_for specifies the redeclaring the defined_for attribute in order to allow the Weld_manufacturing_definition to point to Weld only. See 4.3.214 for the application assertion.

4.2.321.2 degree_of_automatizations

The degree_of_automatizations specifies the automatization status of the weld manufacturing that indicates the welding is manual, semi-automatic or automatic, etc.

The value of the degree_of_automatizations shall be one of the following:

- automatic;
- manual;
- mechanized;
- robotic;
- self run;
- user defined.

NOTE See 4.2.321.2.1 - 4.2.321.2.6 for the definition of each allowable value for degree_of_automatizations.

4.2.321.2.1

automatic

welding performed by using equipment so designed that welding progresses continuously without constant operating by a worker

4.2.321.2.2

manual

welding where welding operation is performed by hand. semi-automatic, welding performed by using equipment that automatically feeds wire while operating the torch manually

4.2.321.2.3

mechanized

welding performed by the motorization and automatization of operations such as parts supply, clamp, feed, rotation, reverse rotation, clamp release, discharge, etc.

4.2.321.2.4

robotic

a type of automatic welding performed by using an industrial robot, which includes robotic arc welding and robotic spot welding, etc.

4.2.321.2.5

self run

welding in which when a covered electrode is set to a simple equipment, the welding thereafter automatically progresses only by ignition of arc

4.2.321.2.6

user defined

user defined welding automatization degree

4.2.321.3 environment

The environment specifies the identifier used to denote whether the weld has been designated to be made in a controlled environment (i.e., the shop) or in a non-controlled environment (i.e., the field). Welds made in the field are usually more difficult to make and less efficiently produced.

The value of the environment shall be one of the following:

— field;

— shop.

NOTE See 4.2.321.3.1 - 4.2.321.3.2 for the definition of each allowable value for environment.

4.2.321.3.1

field

the non-controlled environment in the open

4.2.321.3.2

shop

the controlled environment in a shop

4.2.321.4 number_of_weld_passes

The number_of_weld_passes specifies a count measuring the number of incremental passes the person making the weld needs to perform in order to archive the final weld thickness. Depending on the weld process used, only a fixed amount of weld filler material may be laid down by

an electrode in a single layer. When welds thicker than this amount are required, multiple passes must be used to build up to the final thickness and to suit the required cross-section of weld.

EXAMPLE 1 If a 12mm fillet weld is required and a 6mm rod is used, then 2 passes would be necessary.

4.2.321.5 penetration

The penetration specifies the identifier used to determine the result of weld whether the weld metal in "full" or "partial" penetration with the base metal of the parts being joined. A full penetration weld is stronger than a partial penetration weld and there are situations and areas of a ship where full penetration welds are required. The most significant factor affecting whether a weld detail is full or partial are weld parameters such as root opening, bevel angles and backing.

The value of the penetration shall be one of the following:

- full;
- partial.

NOTE See 4.2.321.5.1 - 4.2.321.5.2 for the definition of each allowable value for penetration.

4.2.321.5.1

full

the weld metal penetrates fully through the base metal of the parts being joined

4.2.321.5.2

partial

the weld metal penetrates only partially through the base metal of the parts being joined

4.2.321.6 position

The position specifies the orientation of the weld with respect of the person performing the welding. The position affects the rate at which weld material can be deposited and factors into weld estimating calculation. All welding positions are indicated by the angle of rotation and the angle of inclination.

The value of the position shall be one of the following:

- flat;
- horizontal;
- overhead;
- other.

— vertical;

NOTE See 4.2.321.6.1 - 4.2.321.6.5 for the definition of each allowable value for position.

4.2.321.6.1

flat

the welding position to weld while facing downward from the top against a joint whose welding axis is almost horizontal. It is used mostly in welding with manual electrodes, for example at the deck from above

4.2.321.6.2

horizontal

the welding position to put bead horizontally against a joint whose welding axis is almost horizontal while the parts being welded are in a vertical position, for example across a bulkhead

4.2.321.6.3

other

other possible position

4.2.321.6.4

vertical

the welding position to put bead vertically from the top or bottom against a joint whose welding axis is almost vertical, both parts being welded are also in vertical position, for example up a bulkhead

4.2.321.6.5

overhead

the welding position to weld while facing upward from the bottom against a joint whose welding axis is almost horizontal, the parts being welded are over the head of the welder, for example welding at the deck from below

Figure 50 shows weld positions.

4.2.321.7 process

The process specifies the process to be used to make the weld. The process currently used in shipbuilding is the electric arc welding that includes slag shielded process such as SMAW, SAW and FCAW, as well as the inert gas shielded process such as GMAW and FCAW.

The value of the process shall be one of the following:

— FCAW;

— GMAW;

— SAW;

- SMAW;
- user defined.

NOTE See 4.2.321.7.1 - 4.2.321.7.5 for the definition of each allowable value for process.

4.2.321.7.1

FCAW

the flux-cored arc welding

4.2.321.7.2

GMAW

the gas metal arc welding

4.2.321.7.3

SAW

the sub-merged arc welding

4.2.321.7.4

SMAW

the manually shielded metal arc welding

4.2.321.7.5

user defined

other user defined non-standard arc welding process

4.2.321.8 sequences

The sequences specifies the welding sequences developed to specify the sequences involving welding of ship structures. To reduce distortion and limit the residual stress in the structure it is important that a correct welding sequence should be utilized throughout the construction. This applies both during the fabrication of units and at erection and joining of units on the building berth. See 4.3.216 for the application assertion.

4.2.321.9 tack_weld_used

The tack_weld_used specifies the indicator as to explain whether or not a tack weld is used. The tack weld holds plates and sections in place after alignment and prior to completion of the continuous fillet and butt weld.

4.2.321.10 torch_vector

The torch_vector specifies the vector of the welding torch, which indicates the direction of the torch which the welder holds. See 4.3.213 for the application assertion.

4.2.321.11 weld_test

The weld_test specifies the test to be done to qualify the weld. The test methods, found weld faults and test results should be given to judge the weld qualification. See 4.3.215 for the application assertion.

4.2.321.12 welding_deposition_sequences

The welding_deposition_sequences specifies the methods of welding deposition used to minimize distortion of welding.

Figure 51 shows different weld methods: backstep and wandering methods of welding.

Figure 52 shows the weld distortion.

The value of the welding_deposition_sequences shall be one of the following:

- progressive;
- backstep;
- wandering;
- user_defined.

NOTE See 4.2.321.12.1 - 4.2.321.12.4 for the definition of each allowable value for welding-deposition_sequences.

4.2.321.12.1

progressive

method in which deposition is carried out so that a welding direction of welder has the same direction as a deposition direction

4.2.321.12.2

backste

method in which deposition is carried out so that a welding direction of welder becomes reverse to a deposition direction

4.2.321.12.3

wandering

method in which both progressive and backstep methods are combined used

4.2.321.12.4

user_defined

a deposition sequence defined by the user

4.2.321.13 welding_procedures

The welding_procedures specifies the text instructions developed to ensure that the proper care, preparation and actions are taken or followed during production welding of the joint. These procedures address such things as pre-head, post-head, edge preparation, cooling time between successive weld passes, etc. See 4.3.212 for the application assertion.

Figure 53 shows weld procedures.

4.2.322 Weld_testing

A Weld_testing contains information that outlining the test methods and test steps that must be followed to ensure that the weld was properly made. Test pieces shall be evaluated according to the acceptance requirements specified for relevant types of imperfections.

The data associated with a Weld_testing are the following:

- test_methods;
- test_results;
- test_sequence.

4.2.322.1 test_methods

The test_methods specifies the methods for welding tests, they are required to enable the soundness of ship weld to be assessed. The various available non-destructive testing methods may be summarized.

The value of the test_methods shall be one of the following:

- dye penetrant;
- magnetic particle;
- radiographic;
- ultrasonic.
- visual examination;

NOTE See 4.2.322.1.1 - 4.2.322.1.5 for the definition of each allowable value for test_methods.

4.2.322.1.1 dye penetrant

a method to show up a surface flaw if it remains after the casting part has been washed following the application of the dye. The dye penetrant test has a small application in ship hull construction, being used for examining for surface cracks in stern frames and other castings. To aid the detection of a surface crack the dye penetrant used is often luminous and is revealed under an ultra-violet light

4.2.322.1.2

magnetic particle

carried out by magnetizing the casting parts, and spreading a fluid of magnetic particles (e.g., iron filings suspended in paraffin) on the surface. Any discontinuity such as a surface crack will show up as the particles will concentrate at this point where there is an alteration in the magnetic field. The magnetic particle test has a small application in ship hull construction, being used for examining for surface cracks in stern frames and other castings

4.2.322.1.3

radiographic

a inspection method the principle of which is simply to subject a material to radiation from one side, and record the radiation emitted from the opposite side. Any obstacle in the path of the radiation will affect the radiation density emitted and may be recorded. As radiation will expose photographic plate, for all practice weld test purpose this is used to record the consistency of the weld metal. Either X-ray or gamma ray devices may be used to provide the source of radiation

4.2.322.1.4

ultrasonic

used increasingly as a tool for locating defects in welds. The technique is particularly useful for locating fine cracks. The principle of ultrasonic depends on the fact that pulses of ultrasonic energy are reflected from any surface which they encounter. Ultrasonic waves traveling through a plate may be reflected from the surface of the metal and also from the surface of any flaws which exist in the metal

4.2.322.1.5

visual examination

a routine procedure executed by sight, and surface defects are soon noticed by the experienced inspector. Incorrect bead shape, high spatter, undercutting, bad stop and start points, incorrect alignment, and surface cracks are all faults which may be observed at the surface. This examination is commonly used in ship hull construction, is made before, during and after welding

Figure 55 shows the welding sequences.

4.2.322.2 test_results

The test_results specifies the results of a weld test. The results assess a weld qualification status in accordance with ISO standards. A weld is approved if the imperfections in the test piece are within the specified limits of ISO standards. Reference should be also made to the corresponding acceptance criteria for non-destructive examination.

The value of the test_results shall be one of the following:

- other;
- pending;
- pre qualified;
- rejected.

NOTE See 4.2.322.2.1 - 4.2.322.2.4 for the definition of each allowable value for test_results.

4.2.322.2.1

other

other possible status

4.2.322.2.2

pending

the weld status is not finally judged, in this case an additional test shall be required in order to assess the quality and integrity of the weld

4.2.322.2.3

rejected

a not approved status which indicates that the imperfections in the test piece exceed the permitted maximum specified

4.2.322.2.4

pre qualified

an approved status which indicates that the imperfections in the test piece are within the specified limits

4.2.322.3 test_sequence

The test_sequence specifies the text instruction to give the necessary test sequence, such as which test should be done at first, which method is necessary for a weld in hull structure, etc. See 4.3.217 for the application assertion.

4.2.323 Welded_joint

A Welded_joint is a type of Structural_part_joint (see 4.2.273) that represents a connection between parts that is made by heating the part edges in the vicinity of the joint to a temperature sufficient to melt the metal while simultaneously depositing additional weld filler material to the joint opening. As the material cool, the parts are permanently bonded together.

The data associated with a Welded_joint are the following:

- joint_type;

— realization.

4.2.323.1 joint_type

The `joint_type` denotes the type of joint represented by a `Welded_joint`.

4.2.323.2 realization

The realization redeclares the realization attribute in order to allow the `Welded_joint` to point to a `Weld` only.

Figure 54 shows different weld types.

4.2.324 Welded_joint_design_definition

A `Weld_design_definition` is a type of `Structural_part_joint_design_definition` (see 4.2.274) that specifies the the design aspects of a connection, such as its characteristics, its shape and class.

The data associated with a `Welded_joint_design_definition` are the following:

- configuration;
- defined_for;
- representations.

4.2.324.1 configuration

The configuration specifies the actual arrangement of the structural parts being joined at the structural part joint. The supported configurations are butt, tee, corner, lap and edge. The required connection configuration should agree with this physical implementation.

The value of the configuration shall be one of the following:

- butt;
- corner;
- cruciform;
- edge;
- flare groove;
- lap ;

- stud;
- surfacing;
- tee;
- user defined.

NOTE See 4.2.324.1.1 - 4.2.324.1.10 for the definition of each allowable value for configuration.

4.2.324.1.1

butt

weld joints in which the base metals lie nearly in the same plane

4.2.324.1.2

corner

weld joints at a corner where two base metals are kept in an L shape making approximately a right angle

4.2.324.1.3

cruciform

weld joints forming a cross

4.2.324.1.4

edge

weld joints at a common end of two base metals with or without partially equal surfaces overlapping. These include e.g. single flange and double flange welding

4.2.324.1.5

flare groove

weld joints whose groove form is constructed from arc to arc or from arc to line

4.2.324.1.6

lap

weld joints formed by partially overlapping base metal. These include fillet, spot, seam weldings, brazing and soldering

4.2.324.1.7

stud

weld joints whose welding is performed by generating arc between the tip of a bolt, bar, etc. and base metal, and pressing it into the resulting molten pool

4.2.324.1.8

surfacing

weld joints in which metal is deposited on a base metal surface, such as cladding by welding, thermal spraying, etc.

4.2.324.1.9

tee

weld joints making approximately a right angle in a T shape formed by putting the end of one plate on the surface of another plate

4.2.324.1.10

user defined

other user defined possible joint configurations

4.2.324.2 defined_for

The `defined_for` specifies the redeclaring the `defined_for` attribute in order to allow the `Welded_design_definition` to point to `Welded_joint` only. See 4.3.219 for the application assertion.

4.2.324.3 representations

The `representations` specifies that the `Welded_joint_design_definition` shall be represented by `Shape_representations`. See 4.3.218 for the application assertion.

4.2.325 Welding_sequence

A `Welding_sequence` defines the production sequence of the `Weld` (see 4.2.318).

The data associated with a `Welding_sequence` are the following:

- `direction`;
- `end_offset`;
- `name`;
- `start_offset`.

4.2.325.1 direction

The `direction` specifies the flag indicating the welding direction. True: from start point to the end point of the `Weld`. False: from the end point to the start point of the `Weld`.

4.2.325.2 end_offset

The `end_offset` specifies the offset from the end point of a `Welding_sequence`.

4.2.325.3 name

The `name` specifies the name of `Welding_sequence`.

4.2.325.4 start_offset

The start_offset specifies the offset from the start point of a Welding_sequence.

4.2.326 Working_Ship

A Working_Ship is a type of ShipType (see 4.2.243) that is a ship which is constructed to perform specific tasks.

The data associated with a Working_Ship are the following:

— has_type.

The has_type specifies the type of workingh ships.

The value of the has_type shall be one of the following:

- Crane Vessel;
- Dredger;
- Drilling Vessel;
- Fire Fighter;
- Fishing Vessel;
- Floating Dock;
- Floating Hotel;
- FPGO;
- FPSO;
- Ice breaker;
- Offshore Supply Vessel;
- Oil Production Vessel;
- Oil Production and Storage Vessel;
- Oil Storage Vessel;
- Pilot Boat;

- Pipe Laying Vessel;
- Pusher;
- Shuttle tanker;
- Supply Vessel;
- Tug;
- Well Stimulation Vessel;
- user defined.

NOTE See 4.2.326.1.1 - 4.2.326.1.25 for the definition of each allowable value for has_type.

4.2.326.1

Crane Vessel

a working ship constructed for lifting purposes

4.2.326.2

Dredger

a working ship constructed for dredging channels or harbour entrances

4.2.326.3

Drilling Vessel

a working ship constructed for drilling purposes

4.2.326.4

Fire Fighter

a working ship constructed for fire fighting purposes

4.2.326.5

Fishing Vessel

a working ship constructed for fishing

4.2.326.6

Floating Dock

a working ship constructed for lifting ships for repair purposes

4.2.326.7

Floating Hotel

a passenger ship constructed for hotel purpose

4.2.326.8

FPGO

a floating platform used for production and storage of gas

4.2.326.9

FPSO

a floating platform used for production and storage of oil

4.2.326.10

Ice breaker

a working ship constructed for breaking ice

4.2.326.11

Offshore Supply Vessel

a working ship constructed for supplying the offshore

4.2.326.12

Oil Production Vessel

a working ship constructed for production of oil

4.2.326.13

Oil Production and Storage Vessel

a working ship constructed for production and storage of oil

4.2.326.14

Oil Storage Vessel

a working ship constructed for storage of oil

4.2.326.15

Pilot Boat

a working ship constructed for carrying the pilot to the ship which he conducts in and out of the harbour

4.2.326.16

Pipe Laying Vessel

a working ship constructed for pipe laying purposes

4.2.326.17

Pusher

a working ship constructed to push other unpropelled ships e.g. a barge

4.2.326.18

Shuttle tanker

a ship equipped for offshore oil loading. It may be used for transporting oil from a platform to shore

4.2.326.19

Supply Vessel

a working ship constructed for supplying purposes e.g. offshore supply vessel

4.2.326.20

Tug

a working ship constructed for towing purposes

4.2.326.21

Well Stimulation Vessel

a ship used for the well Stimulation

4.2.326.22

user defined

can be any ship type which is not in the enumeration list. Details should be found in the ship_type_description

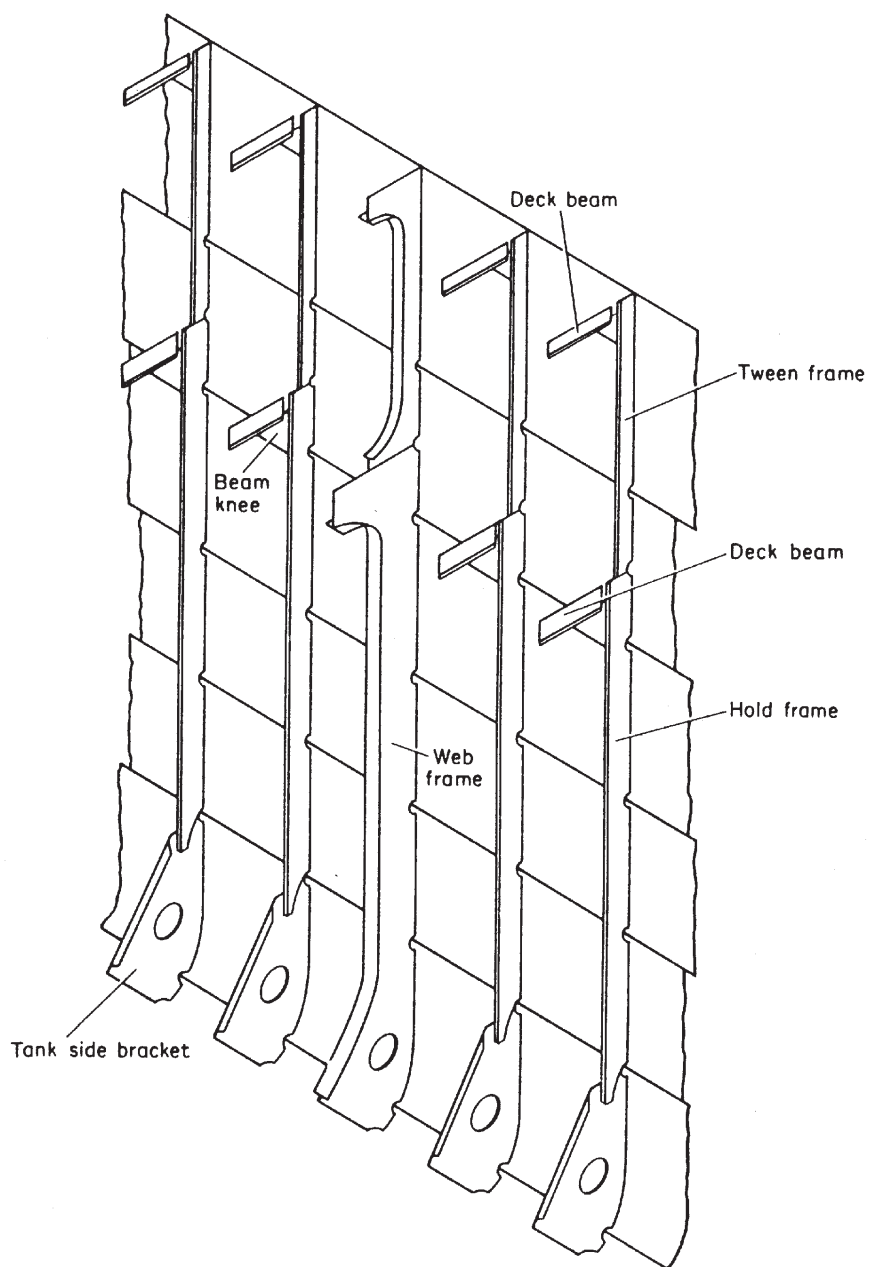


Figure 40 – A SHELL

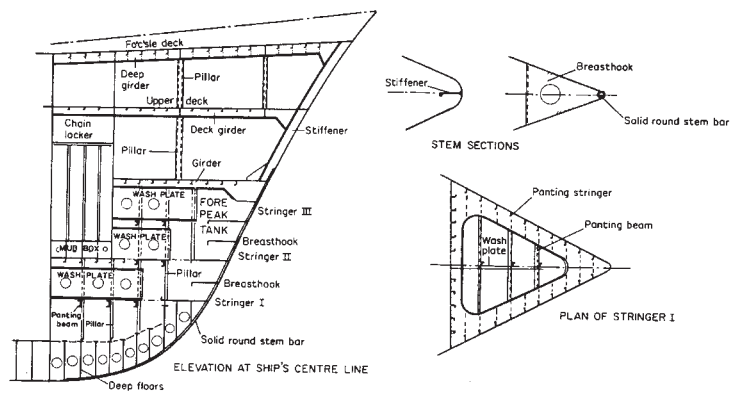


Figure 41 – A 2-dimensional FORE SHIP

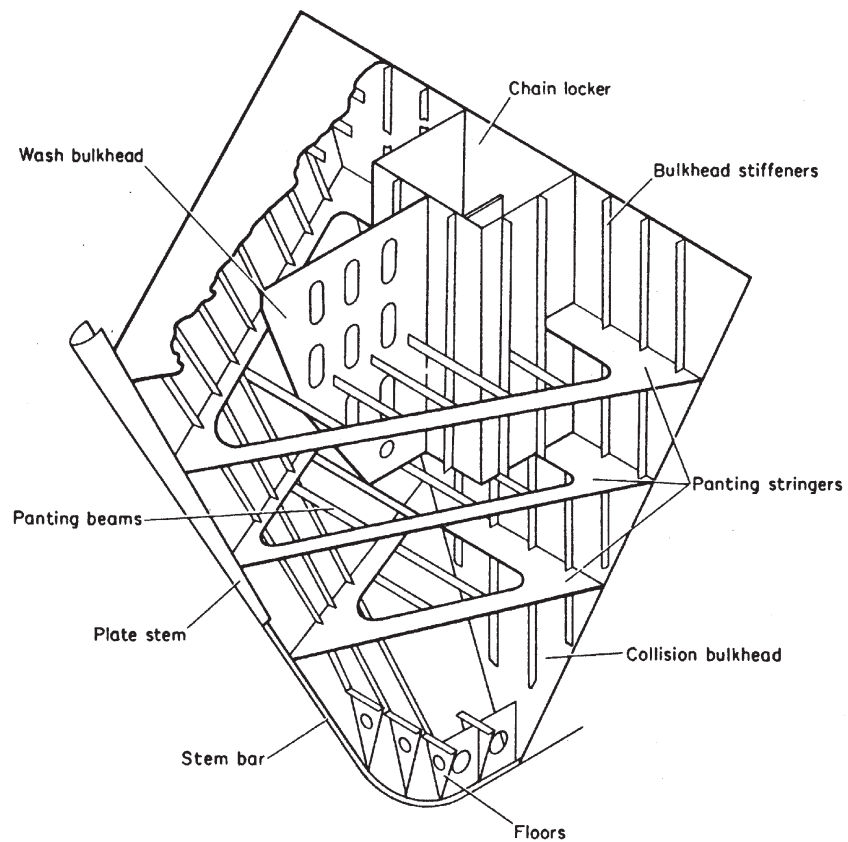


Figure 42 – The FORE SHIP in a 3-dimensional view

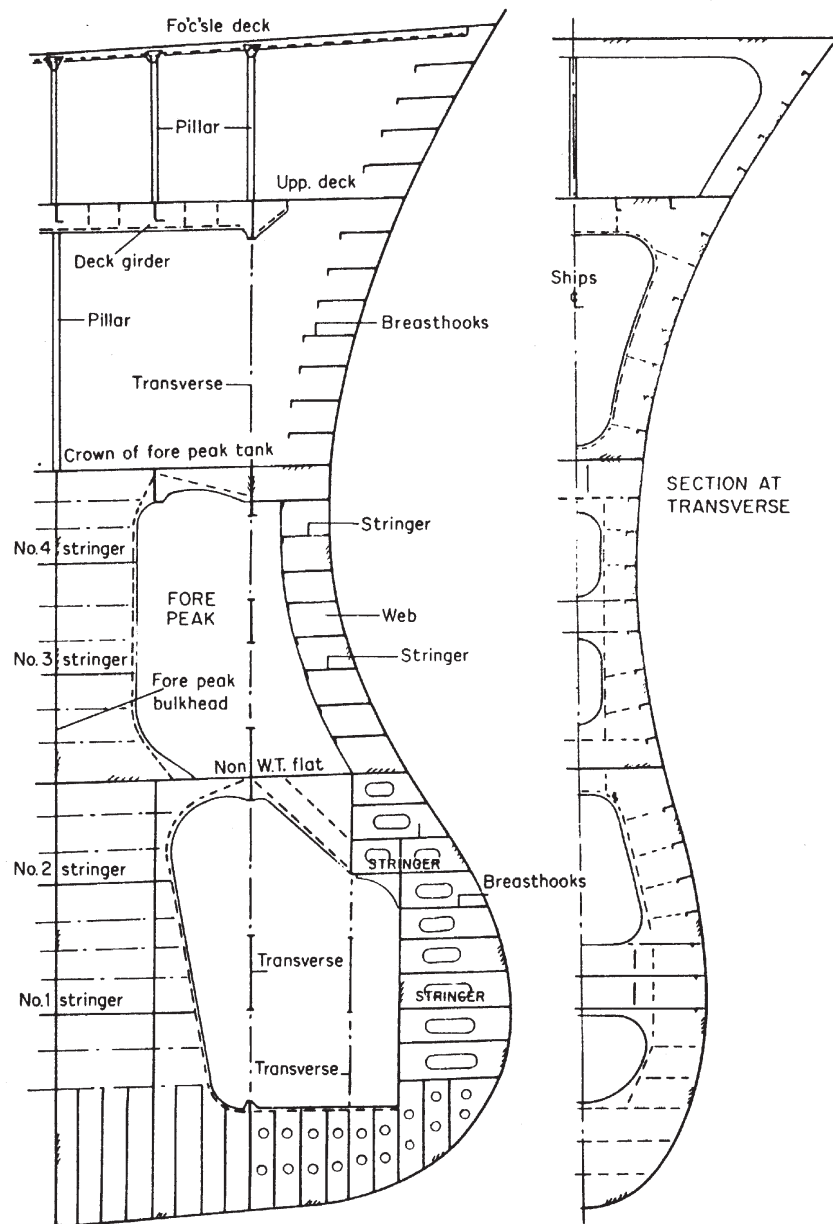


Figure 43 – A bow

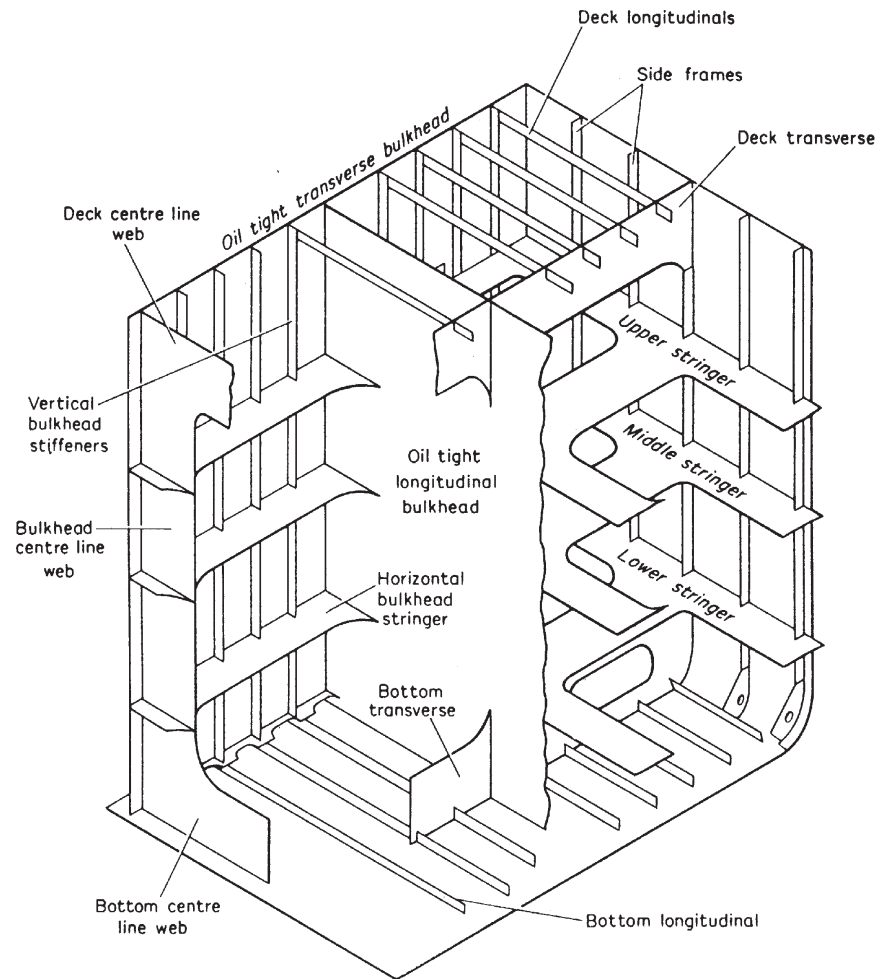


Figure 44 – A TANK

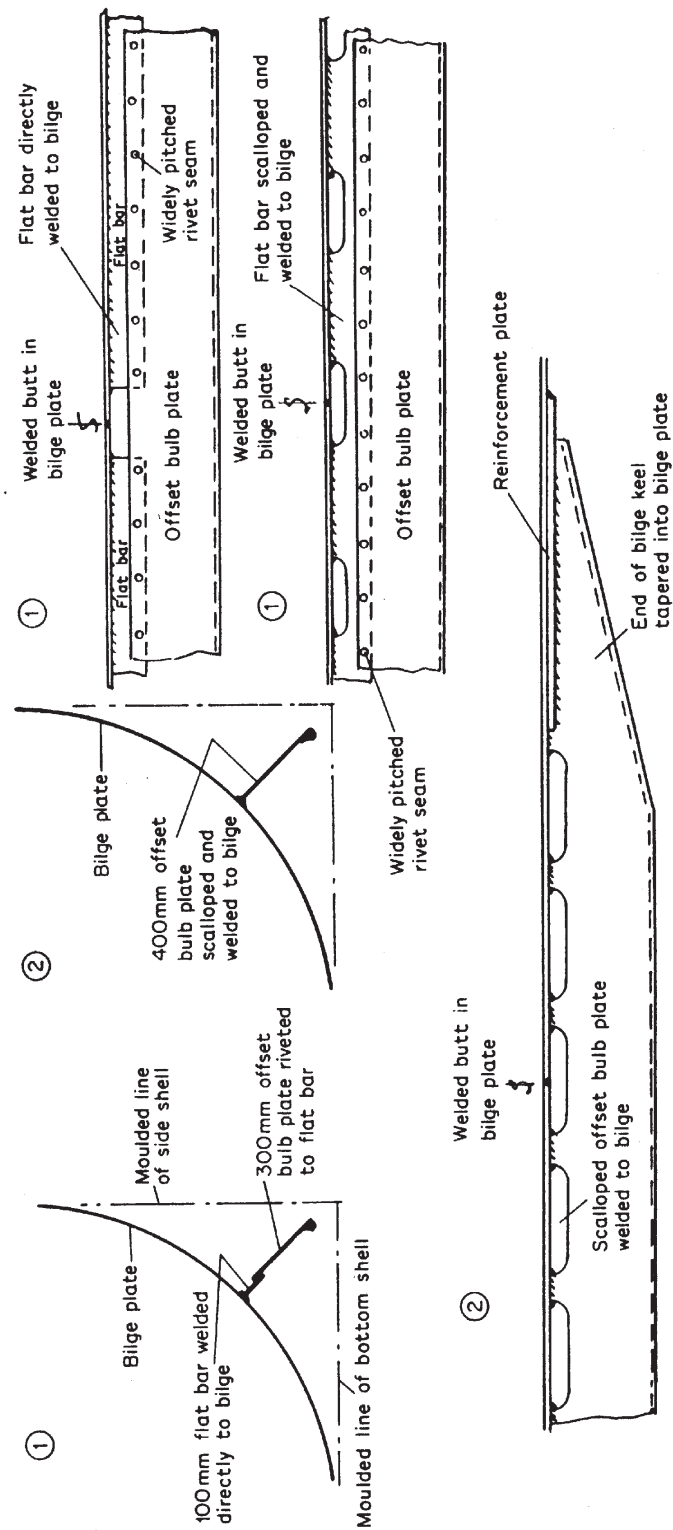


Figure 45 - A KEEL

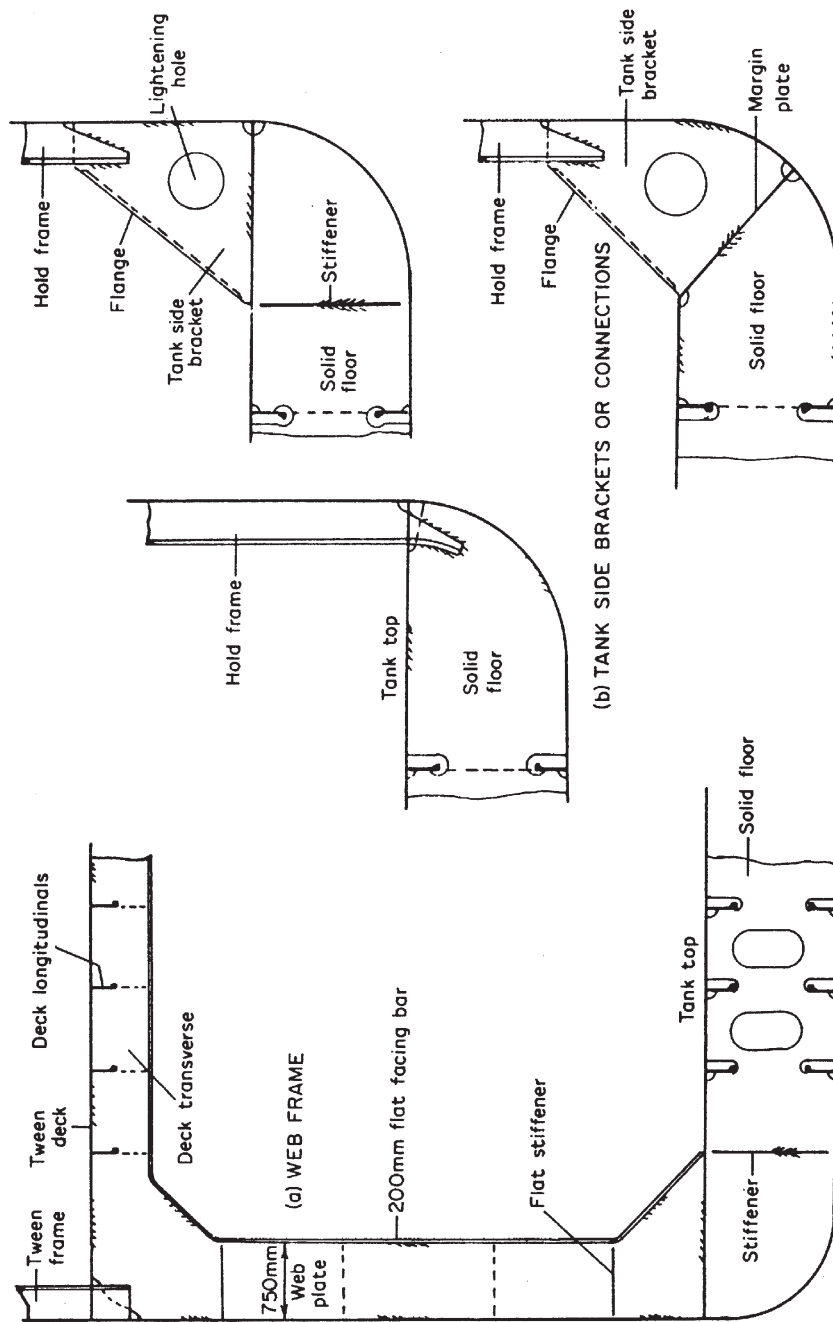


Figure 46 – A WEB FRAME

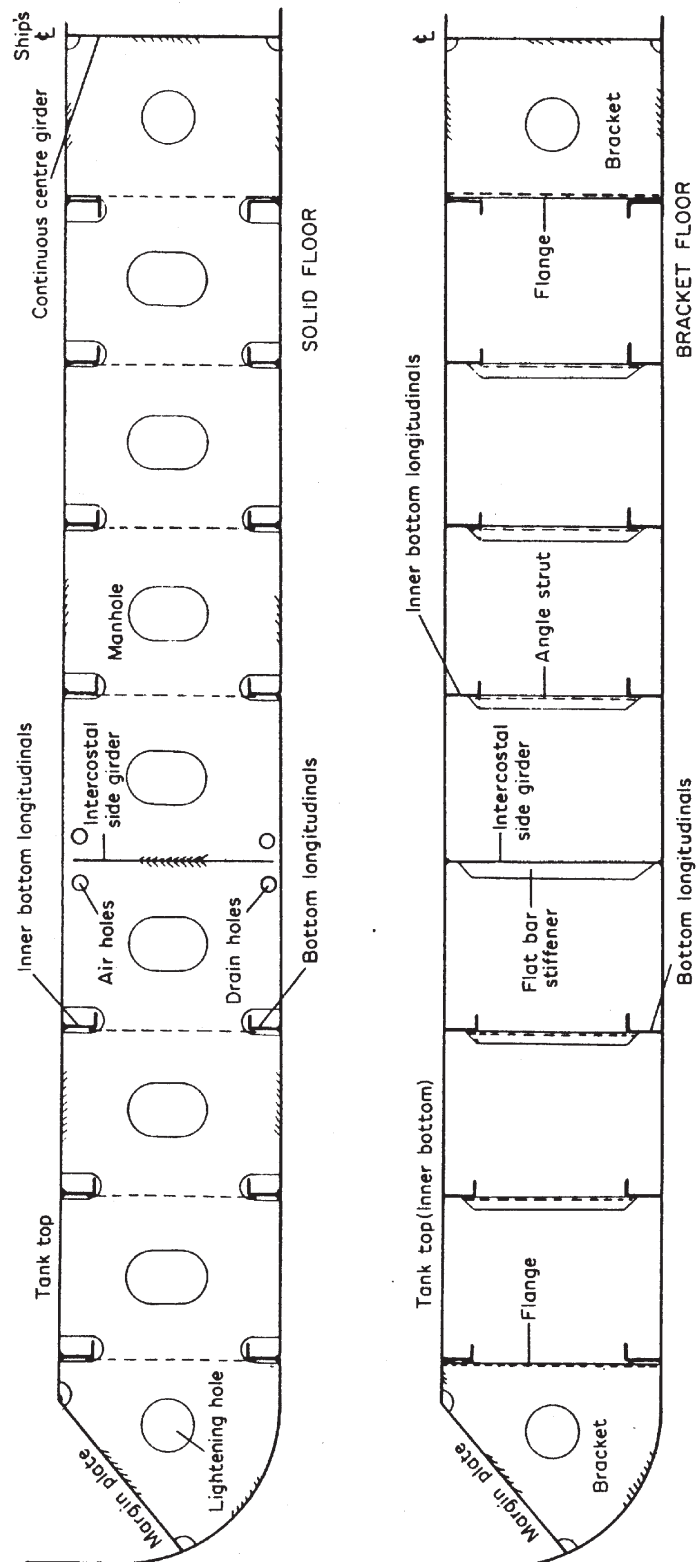


Figure 47 – A LONGITUDINAL GIRDER

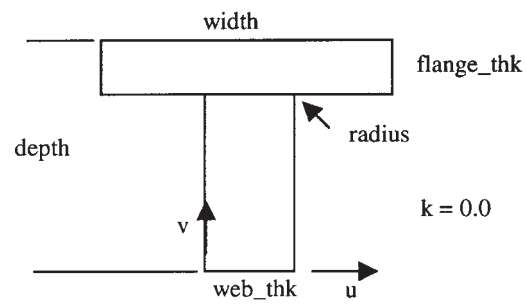
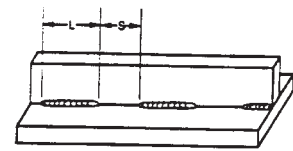
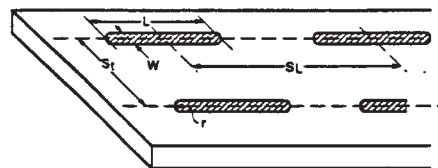


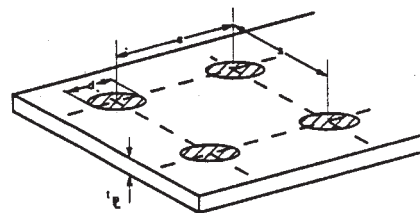
Figure 48 – A T_bar_cross_section



EXAMPLE OF INTERMITTENT WELDING



SLOT WELD ATTRIBUTES



PLUG WELD ATTRIBUTES

Figure 49 – Examples of welds

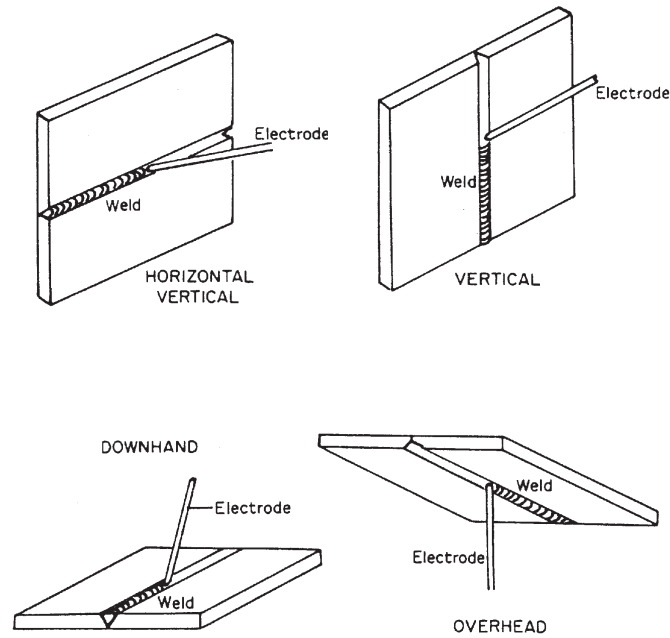


Figure 50 – Weld positions

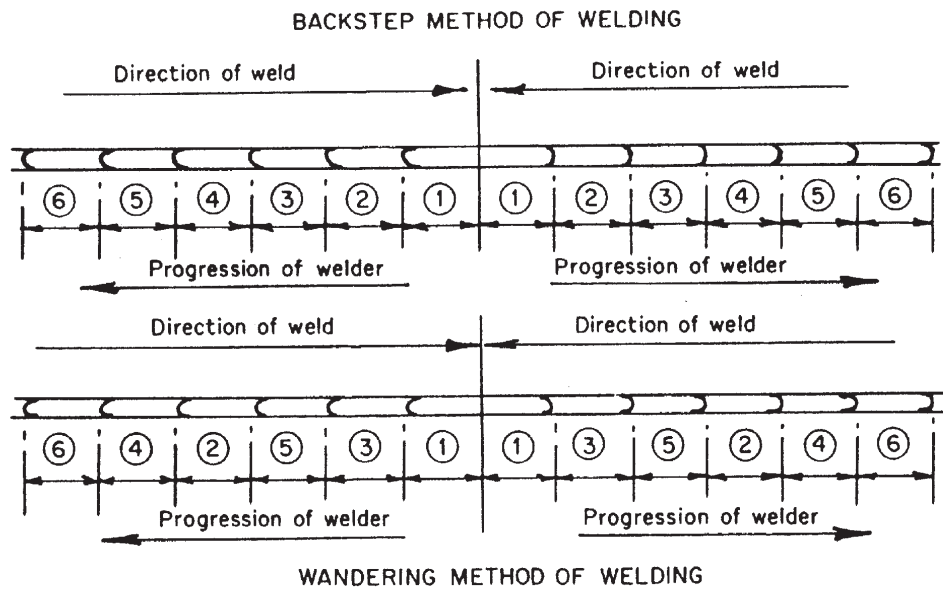


Figure 51 – Weld methods

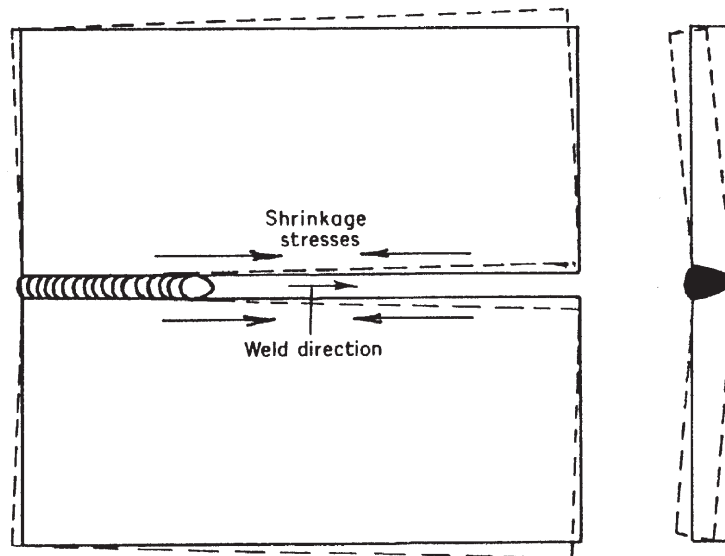


Figure 52 – Backstep and wandering methods of welding

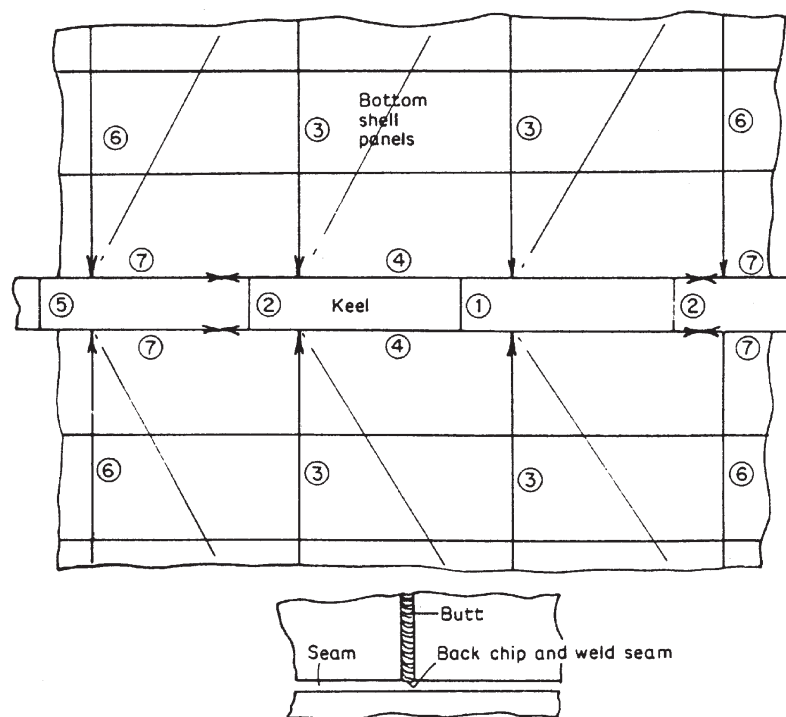






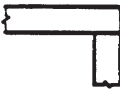





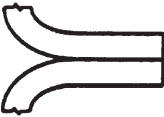




Figure 53 – Weld procedures

JOINT CONFIGURATION		BUTT AND FILLET WELD TYPES		
		Single	Double	
Butt				
Tee				
Corner				
Lap				
Edge				

REPRESENTATIVE JOINT AND WELD TYPES

Figure 54 – Weld types

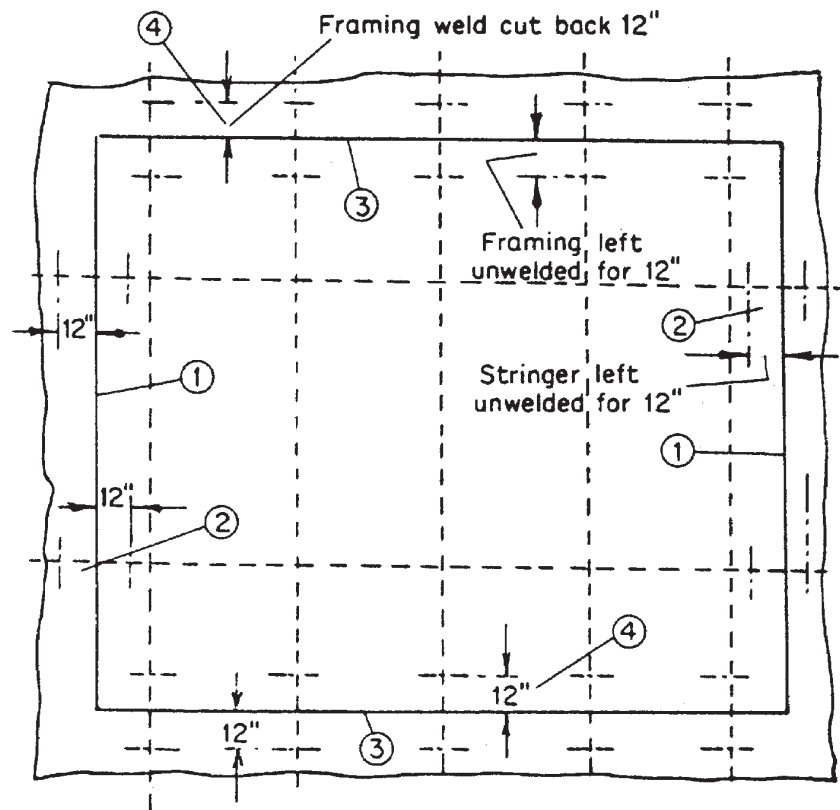


Figure 55 – Weld sequences

4.3 Application assertions

This subclause specifies the application assertions for the Ship Structures application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

4.3.1 Approval_history to Approval_event

Each Approval_history has approvals defined by one up to many Approval_event objects. Approval_event defines approval_reference for exactly one Approval_history.

4.3.2 Approval_history to Definition

Each Approval_history has subject defined by exactly one Definition. A Definition defines the subject for an Approval_history.

4.3.3 Assembly_bounding_box to Cartesian_point

Each Assembly_bounding_box has point_max defined by exactly one Cartesian_point. A Cartesian_point defines the point_max for an Assembly_bounding_box.

Each Assembly_bounding_box has point_min defined by exactly one Cartesian_point. A Cartesian_point defines the point_min for an Assembly_bounding_box.

4.3.4 Assembly_manufacturing_definition to Assembly

Each Assembly_manufacturing_definition has defined_for defined by one up to many Assembly objects. A set of Assembly objects defines the defined_for for an Assembly_manufacturing_definition.

4.3.5 Assembly_manufacturing_definition to Assembly_manufacturing_position

Each Assembly_manufacturing_definition has assembly_positions defined by zero up to many Assembly_manufacturing_position objects. A set of Assembly_manufacturing_position objects defines the assembly_positions for an Assembly_manufacturing_definition.

4.3.6 Assembly_manufacturing_definition to Cartesian_point

Each Assembly_manufacturing_definition has centre_of_gravity defined by exactly one Cartesian_point. A Cartesian_point defines the centre_of_gravity for an Assembly_manufacturing_definition.

4.3.7 Assembly_manufacturing_definition to Date_and_time

Each Assembly_manufacturing_definition has production_date defined by exactly one Date_and_time. A Date_and_time defines the production_date for an Assembly_manufacturing_definition.

4.3.8 Assembly_manufacturing_definition to Document_reference

Each Assembly_manufacturing_definition has assembly_drawing defined by exactly one Document_reference. A Document_reference defines the assembly_drawing for an Assembly_manufacturing_definition.

4.3.9 Assembly_manufacturing_position to Assembly_bounding_box

Each Assembly_manufacturing_position has bounding_box defined by exactly one Assembly_bounding_box. An Assembly_bounding_box defines the bounding_box for an Assembly_manufacturing_position.

4.3.10 Assembly_manufacturing_position to Bounded_curve

Each Assembly_manufacturing_position has assembly_footprint defined by exactly one Bounded_curve. A Bounded_curve defines the assembly_footprint for an Assembly_manufacturing_position.

4.3.11 Assembly_manufacturing_position to Direction

Each Assembly_manufacturing_position has orientation defined by exactly one Direction. A Direction defines the orientation for an Assembly_manufacturing_position.

4.3.12 Assembly_relationship to Assembly

Each Assembly_relationship has item_1 defined by exactly one Assembly. An Assembly defines the item_1 for an Assembly_relationship.

4.3.13 Change_definition to Change

Each Change_definition has defined_for defined by one up to many Change objects. A set of Change objects defines the defined_for for a Change_definition.

4.3.14 Change_definition to Date_and_time

Each Change_definition has date_time defined by exactly one Date_and_time. A Date_and_time defines the date_time for a Change_definition.

4.3.15 Change_definition to Person_and_organisation

Each Change_definition has author defined by exactly one Person_and_organisation. A Person_and_organisation defines the author for a Change_definition.

4.3.16 Change_impact to Versionable_object_change_event

Each Change_impact has impact defined by one up to many Versionable_object_change_event objects. A set of Versionable_object_change_event objects defines the impact for a Change_impact.

4.3.17 Change_plan to Change_impact

Each Change_plan has planned_impact defined by exactly one Change_impact. A Change_impact defines the planned_impact for a Change_plan.

4.3.18 Change_plan to Change_request

Each Change_plan has chosen_solution_for defined by exactly one Change_request. A Change_request defines the chosen_solution_for for a Change_plan.

4.3.19 Change_plan to Check

Each Change_plan has checks defined by zero up to many Check objects. A set of Check objects defines the checks for a Change_plan.

4.3.20 Change_realization to Change_impact

Each Change_realization has impact defined by exactly one Change_impact. A Change_impact defines the impact for a Change_realization.

4.3.21 Change_realization to Change_plan

Each Change_realization has realization_of defined by exactly one Change_plan. A Change_plan defines the realization_of for a Change_realization.

4.3.22 Change_realization to Check

Each Change_realization has checks defined by zero up to many Check objects. A set of Check objects defines the checks for a Change_realization.

4.3.23 Change_request to Change_impact

Each Change_request has solution_alternatives defined by zero up to many Change_impact objects. A set of Change_impact objects defines the solution_alternatives for a Change_request.

4.3.24 Change_request to Person_and_organisation

Each Change_request has addressee defined by zero or one Person_and_organisation. A Person_and_organisation defines the addressee for a Change_request.

Each Change_request has initiator defined by exactly one Person_and_organisation. A Person_and_organisation defines the initiator for a Change_request.

4.3.25 Class_and_statutory_designation to Class_notation

Each Class_and_statutory_designation has class defined by exactly one Class_notation. A Class_notation defines the class for a Class_and_statutory_designation.

4.3.26 Class_and_statutory_designation to Regulation

Each Class_and_statutory_designation has statutory defined by exactly one Regulation. A Regulation defines the statutory for a Class_and_statutory_designation.

4.3.27 Class_bsu to Supplier_bsu

Each Class_bsu has delivered_by defined by exactly one Supplier_bsu. A Supplier_bsu defines the delivered_by for a Class_bsu.

4.3.28 Class_notation to Organisation

Each Class_notation has class_society defined by exactly one Organisation. An Organisation defines the class_society for a Class_notation.

4.3.29 Corner_cutout_boundary_relationship to Corner_cutout

Each Corner_cutout_boundary_relationship has item_1 defined by exactly one Corner_cutout. A Corner_cutout defines the item_1 for a Corner_cutout_boundary_relationship.

4.3.30 Corrugated_part_approval to Corrugated_part_design_definition

Each Corrugated_part_approval has subject defined by exactly one Corrugated_part_design_definition. A Corrugated_part_design_definition defines the subject for a Corrugated_part_approval.

4.3.31 Corrugated_part_design_definition to Corrugated_part

Each Corrugated_part_design_definition has defined_for defined by one up to many Corrugated_part objects. A set of Corrugated_part objects defines the defined_for for a Corrugated_part_design_definition.

4.3.32 Corrugated_part_design_definition to Corrugated_structure_boundary_relationship

Each Corrugated_part_design_definition has border defined by one up to many Corrugated_structure_boundary_relationship objects. A list of Corrugated_structure_boundary_relationship objects defines the border for a Corrugated_part_design_definition.

4.3.33 Corrugated_part_design_definition to Corrugation

Each Corrugated_part_design_definition has shape_aspect defined by exactly one Corrugation. A Corrugation defines the shape_aspect for a Corrugated_part_design_definition.

4.3.34 Corrugated_part_design_definition to Structural_system_adjacency_relationship

Each Corrugated_part_design_definition has border defined by one up to many Structural_system_adjacency_relationship objects. A list of Structural_system_adjacency_relationship objects defines the border for a Corrugated_part_design_definition.

4.3.35 Corrugated_part_design_definition to Surface

Each Corrugated_part_design_definition has border defined by one up to many Surface objects. A list of Surface objects defines the border for a Corrugated_part_design_definition.

4.3.36 Corrugated_part_rejection to Corrugated_part_design_definition

Each Corrugated_part_rejection has subject defined by exactly one Corrugated_part_design_definition. A Corrugated_part_design_definition defines the subject for a Corrugated_part_rejection.

4.3.37 Corrugated_structure_design_definition to Corrugated_structure

Each Corrugated_structure_design_definition has defined_for defined by one up to many Corrugated_structure objects. A set of Corrugated_structure objects defines the defined_for for a Corrugated_structure_design_definition.

4.3.38 Corrugated_structure_functional_definition to Corrugated_structure

Each Corrugated_structure_functional_definition has defined_for defined by one up to many Corrugated_structure objects. A set of Corrugated_structure objects defines the defined_for for a Corrugated_structure_functional_definition.

4.3.39 Corrugated_structure_relationship to Corrugated_structure

Each Corrugated_structure_relationship has item_1 defined by exactly one Corrugated_structure. A Corrugated_structure defines the item_1 for a Corrugated_structure_relationship.

4.3.40 Definable_object to Global_id

Each Definable_object has id defined by exactly one Global_id. A Global_id defines the id for a Definable_object.

4.3.41 Definition to Definable_object

Each Definition has defined_for defined by one up to many Definable_object objects. Definable_object defines definitions for zero up to many Definitions.

4.3.42 Definition to Global_id

Each Definition has id defined by exactly one Global_id. A Global_id defines the id for a Definition.

4.3.43 Definition_relationship to Definition

Each Definition_relationship has definition_1 defined by exactly one Definition. A Definition defines the definition_1 for a Definition_relationship.

Each Definition_relationship has definition_2 defined by exactly one Definition. A Definition defines the definition_2 for a Definition_relationship.

4.3.44 Derived_unit to Derived_unit_element

Each Derived_unit has elements defined by one up to many Derived_unit_element objects. A set of Derived_unit_element objects defines the elements for a Derived_unit.

4.3.45 Derived_unit_element to Named_unit

Each Derived_unit_element has unit defined by exactly one Named_unit. A Named_unit defines the unit for a Derived_unit_element.

4.3.46 Design_definition to Representation

Each Design_definition has representations defined by zero up to many Representation objects. A set of Representation objects defines the representations for a Design_definition.

4.3.47 Design_load to Ship

Each Design_load has defined_for defined by one up to many Ship objects. A set of Ship objects defines the defined_for for a Design_load.

4.3.48 Design_requirement to Document_reference

Each Design_requirement has specification defined by zero up to many Document_reference objects. A set of Document_reference objects defines the specification for a Design_requirement.

4.3.49 Design_still_water_shear_force to Design_SWSF_values

Each Design_still_water_shear_force has table_of_shear_force_values defined by one up to many Design_SWSF_values objects. A set of Design_SWSF_values objects defines the table_of_shear_force_values for a Design_still_water_shear_force.

4.3.50 Design_SWSF_values to Spacing_position

Each Design_SWSF_values has position defined by exactly one Spacing_position. A Spacing_position defines the position for a Design_SWSF_values.

4.3.51 Document_usage_constraint to Document

Each Document_usage_constraint has source defined by exactly one Document. Document defines the document_subset for a Document_usage_constraint.

4.3.52 Drain_hole_cutout_design_definition to Edge_cutout

Each Drain_hole_cutout_design_definition has defined_for defined by one up to many Edge_cutout objects. A set of Edge_cutout objects defines the defined_for for a Drain_hole_cutout_design_definition.

4.3.53 Edge_cutout_functional_definition to Edge_cutout

Each Edge_cutout_functional_definition has defined_for defined by one up to many Edge_cutout objects. A set of Edge_cutout objects defines the defined_for for an Edge_cutout_functional_definition.

4.3.54 Edge_feature_functional_definition to Edge_feature

Each Edge_feature_functional_definition has defined_for defined by one up to many Edge_feature objects. A set of Edge_feature objects defines the defined_for for an Edge_feature_functional_definition.

4.3.55 Envisaged_version_creation to Versionable_object

Each Envisaged_version_creation has base defined by zero up to many Versionable_object objects. A set of Versionable_object objects defines the base for an Envisaged_version_creation.

4.3.56 Event to Date_and_time

Each Event has caused_when defined by exactly one Date_and_time. A Date_and_time defines the caused_when for an Event.

4.3.57 Event to Person_and_organisation

Each Event has caused_by defined by exactly one Person_and_organisation. A Person_and_organisation defines the caused_by for an Event.

4.3.58 Explicit_feature_design_definition to Shape_representation

Each Explicit_feature_design_definition has representations defined by one up to one Shape_representation objects. A set of Shape_representation objects defines the representations for an Explicit_feature_design_definition.

4.3.59 Explicit_profile_cross_section to Axis2_placement_2d

Each Explicit_profile_cross_section has local_coordinate_system defined by exactly one Axis2_placement_2d. An Axis2_placement_2d defines the local_coordinate_system for an Explicit_profile_cross_section.

4.3.60 Explicit_profile_cross_section to Bounded_curve

Each Explicit_profile_cross_section has cross_section_geometry defined by exactly one Bounded_curve. A Bounded_curve defines the cross_section_geometry for an Explicit_profile_cross_section.

4.3.61 Extended_conversion_based_unit to Extended_measure_with_unit

Each Extended_conversion_based_unit has conversion_factor defined by exactly one Extended_measure_with_unit. An Extended_measure_with_unit defines the conversion_factor for an Extended_conversion_based_unit.

4.3.62 Extended_global_uncertainty_assigned_context to Extended_uncertainty_measure_with_unit

Each Extended_global_uncertainty_assigned_context has uncertainty defined by one up to many Extended_uncertainty_measure_with_unit objects. A set of Extended_uncertainty_measure_with_unit objects defines the uncertainty for an Extended_global_uncertainty_assigned_context.

4.3.63 External_instance_reference to Global_id

Each External_instance_reference has target_guid defined by exactly one Global_id. A Global_id defines the target_guid for an External_instance_reference.

4.3.64 Feature_design_definition to Feature

Each Feature_design_definition has defined_for defined by one up to many Feature objects. A set of Feature objects defines the defined_for for a Feature_design_definition.

4.3.65 Feature_relationship to Feature

Each Feature_relationship has item_1 defined by exactly one Feature. A Feature defines the item_1 for a Feature_relationship.

4.3.66 Free_form_interior_cutout_design_definition to Curve

Each Free_form_interior_cutout_design_definition has bounding_curve defined by exactly one Curve. A Curve defines the bounding_curve for a Free_form_interior_cutout_design_definition.

4.3.67 Freeboard_characteristics to Date_and_time

Each Freeboard_characteristics has date_freeboard_assigned defined by exactly one Date_and_time. A Date_and_time defines the date_freeboard_assigned for a Freeboard_characteristics.

4.3.68 Freeboard_characteristics to Loadline

Each Freeboard_characteristics has applicable_loadline defined by exactly one Loadline. A Loadline defines the applicable_loadline for a Freeboard_characteristics.

4.3.69 Freeboard_characteristics to Organisation

Each Freeboard_characteristics has freeboard_assigned_by defined by exactly one Organisation. An Organisation defines the freeboard_assigned_by for a Freeboard_characteristics.

4.3.70 General_characteristics_definition to Ship

Each General_characteristics_definition has defined_for defined by one up to many Ship objects. A set of Ship objects defines the defined_for for a General_characteristics_definition.

4.3.71 Homogeneous_ship_material_property to Part

Each Homogeneous_ship_material_property has defined_for defined by one up to many Part objects. A set of Part objects defines the defined_for for a Homogeneous_ship_material_property.

4.3.72 Hull_cross_section_design_definition to Flare_area

Each Hull_cross_section_design_definition has flare_area_buckling defined by exactly one Flare_area. A Flare_area defines the flare_area_buckling for a Hull_cross_section_design_definition.

4.3.73 Hull_cross_section_design_definition to Hull_cross_section

Each Hull_cross_section_design_definition has defined_for defined by one up to one Hull_cross_section objects. A set of Hull_cross_section objects defines the defined_for for a Hull_cross_section_design_definition.

4.3.74 Hull_cross_section_result to Hull_cross_section_design_definition

Each Hull_cross_section_result has subject defined by exactly one Hull_cross_section_design_definition. A Hull_cross_section_design_definition defines the subject for a Hull_cross_section_result.

4.3.75 Interior_cutout_design_definition to Interior_cutout

Each Interior_cutout_design_definition has defined_for defined by one up to many Interior_cutout objects. A set of Interior_cutout objects defines the defined_for for an Interior_cutout_design_definition.

4.3.76 Interior_cutout_functional_definition to Interior_cutout

Each Interior_cutout_functional_definition has defined_for defined by one up to many Interior_cutout objects. A set of Interior_cutout objects defines the defined_for for an Interior_cutout_functional_definition.

4.3.77 Item to External_reference

Each Item has documentation defined by zero up to many External_reference objects. A set of External_reference objects defines the documentation for an Item.

4.3.78 Item to Ship

Each Item has ship_context defined by zero or one Ship. Ship defines ship_items for zero up to many Items.

4.3.79 Item_relationship to External_instance_reference

Each Item_relationship has external_item_1 defined by zero or one External_instance_reference. An External_instance_reference defines the external_item_1 for an Item_relationship.

Each Item_relationship has external_item_2 defined by zero or one External_instance_reference. An External_instance_reference defines the external_item_2 for an Item_relationship.

4.3.80 Item_relationship to Item

Each Item_relationship has item_1 defined by zero or one Item. An Item defines the item_1 for an Item_relationship.

Each Item_relationship has item_2 defined by zero or one Item. An Item defines the item_2 for an Item_relationship.

4.3.81 Item_structure to External_instance_reference

Each Item_structure has external_items defined by zero up to many External_instance_reference objects. A set of External_instance_reference objects defines the external_items for an Item_structure.

Each Item_structure has external_relationships defined by zero up to many External_instance_reference objects. A set of External_instance_reference objects defines the external_relationships for an Item_structure.

4.3.82 Item_structure to Item

Each Item_structure has items defined by zero up to many Item objects. A set of Item objects defines the items for an Item_structure.

4.3.83 Item_structure to Item_relationship

Each Item_structure has relationships defined by zero up to many Item_relationship objects. A set of Item_relationship objects defines the relationships for an Item_structure.

4.3.84 Library_definition to Library_element_reference

Each Library_definition has source defined by exactly one Library_element_reference. A Library_element_reference defines the source for a Library_definition.

4.3.85 Library_element_reference to Class_bsu

Each Library_element_reference has library_identifier defined by exactly one Class_bsu. A Class_bsu defines the library_identifier for a Library_element_reference.

4.3.86 Library_element_reference to Property_value

Each Library_element_reference has property_value_pairs defined by zero up to many Property_value objects. A set of Property_value objects defines the property_value_pairs for a Library_element_reference.

4.3.87 Lightship_definition to Cartesian_point

Each Lightship_definition has lightship_centre_of_gravity defined by exactly one Cartesian_point. A Cartesian_point defines the lightship_centre_of_gravity for a Lightship_definition.

4.3.88 Lightship_definition to Lightship_weight_item

Each Lightship_definition has lightship_items defined by zero up to many Lightship_weight_item objects. A set of Lightship_weight_item objects defines the lightship_items for a Lightship_definition.

4.3.89 Lightship_definition to Ship

Each Lightship_definition has defined_for defined by one up to one Ship objects. A set of Ship objects defines the defined_for for a Lightship_definition.

4.3.90 Local_co_ordinate_system_with_position_reference to Cartesian_point

Each Local_co_ordinate_system_with_position_reference has location defined by exactly one Cartesian_point. A Cartesian_point defines the location for a Local_co_ordinate_system_with_position_reference.

4.3.91 Local_co_ordinate_system_with_position_reference to Spacing_position

Each Local_co_ordinate_system_with_position_reference has longitudinal_ref defined by exactly one Spacing_position. A Spacing_position defines the longitudinal_ref for a Local_co_ordinate_system_with_position_reference.

Each Local_co_ordinate_system_with_position_reference has transversal_ref defined by exactly one Spacing_position. A Spacing_position defines the transversal_ref for a Local_co_ordinate_system_with_position_reference.

Each Local_co_ordinate_system_with_position_reference has vertical_ref defined by exactly one Spacing_position. A Spacing_position defines the vertical_ref for a Local_co_ordinate_system_with_position_reference.

4.3.92 Longitudinal_table to Longitudinal_position

Each Longitudinal_table has spacing_table_representations defined by zero up to many Longitudinal_position objects. A list of Longitudinal_position objects defines the spacing_table_representations for a Longitudinal_table.

4.3.93 Manufacturing_definition to Representation

Each Manufacturing_definition has representations defined by zero up to many Representation objects. A set of Representation objects defines the representations for a Manufacturing_definition.

4.3.94 Maximum_permmissible_still_water_bending_moment to Maximum_permmissible_SWBM_values

Each Maximum_permmissible_still_water_bending_moment has table_of_max_values defined by one up to many Maximum_permmissible_SWBM_values objects. A set of Maximum_permmissible_SWBM_values objects defines the table_of_max_values for a Maximum_permmissible_still_water_bending_moment.

4.3.95 Maximum_permmissible_still_water_shear_force to Maximum_permmissible_SWSF_values

Each Maximum_permmissible_still_water_shear_force has table_of_max_shear_force_values defined by one up to many Maximum_permmissible_SWSF_values objects. A set of Maximum_permmissible_SWSF_values objects defines the table_of_max_shear_force_values for a Maximum_permmissible_still_water_shear_force.

4.3.96 Maximum_permmissible_SWBM_values to Spacing_position

Each Maximum_permmissible_SWBM_values has position defined by exactly one Spacing_position. A Spacing_position defines the position for a Maximum_permmissible_SWBM_values.

4.3.97 Maximum_permmissible_SWSF_values to Spacing_position

Each Maximum_permmissible_SWSF_values has position defined by exactly one Spacing_position. A Spacing_position defines the position for a Maximum_permmissible_SWSF_values.

4.3.98 Moment_3d to Cartesian_point

Each Moment_3d has origin defined by exactly one Cartesian_point. A Cartesian_point defines the origin for a Moment_3d.

4.3.99 Named_unit to Dimensional_exponents

Each Named_unit has dimensions defined by exactly one Dimensional_exponents. A Dimensional_exponents defines the dimensions for a Named_unit.

4.3.100 Owner_designation to Organisation

Each Owner_designation has managing_company defined by exactly one Organisation. An Organisation defines the managing_company for an Owner_designation.

Each Owner_designation has ordering_company defined by exactly one Organisation. An Organisation defines the ordering_company for an Owner_designation.

Each Owner_designation has owning_company defined by exactly one Organisation. An Organisation defines the owning_company for an Owner_designation.

4.3.101 Panel_system_curve_boundary to Bounded_curve

Each Panel_system_curve_boundary has curve defined by exactly one Bounded_curve. A Bounded_curve defines the curve for a Panel_system_curve_boundary.

4.3.102 Panel_system_design_definition to Panel_system

Each Panel_system_design_definition has defined_for defined by one up to many Panel_system objects. A set of Panel_system objects defines the defined_for for a Panel_system_design_definition.

4.3.103 Panel_system_functional_definition to Panel_system

Each Panel_system_functional_definition has defined_for defined by one up to many Panel_system objects. A set of Panel_system objects defines the defined_for for a Panel_system_functional_definition.

4.3.104 Panel_system_plane_boundary to Plane

Each Panel_system_plane_boundary has plane defined by exactly one Plane. A Plane defines the plane for a Panel_system_plane_boundary.

4.3.105 Panel_system_relationship to Panel_system

Each Panel_system_relationship has item_1 defined by exactly one Panel_system. A Panel_system defines the item_1 for a Panel_system_relationship.

4.3.106 Part_edge_cutout_design_definition to Edge_cutout

Each Part_edge_cutout_design_definition has defined_for defined by one up to many Edge_cutout objects. A set of Edge_cutout objects defines the defined_for for a Part_edge_cutout_design_definition.

4.3.107 Part_relationship to Part

Each Part_relationship has item_1 defined by exactly one Part. A Part defines the item_1 for a Part_relationship.

4.3.108 Plate_approval to Plate_design_definition

Each Plate_approval has subject defined by exactly one Plate_design_definition. A Plate_design_definition defines the subject for a Plate_approval.

4.3.109 Plate_boundary_relationship to Seam

Each Plate_boundary_relationship has item_2 defined by exactly one Seam. A Seam defines the item_2 for a Plate_boundary_relationship.

4.3.110 Plate_design_definition to Bounded_curve

Each Plate_design_definition has border defined by one up to many Bounded_curve objects. A set of Bounded_curve objects defines the border for a Plate_design_definition.

4.3.111 Plate_design_definition to External_instance_reference

Each Plate_design_definition has moulded_surface defined by exactly one External_instance_reference object. A External_instance_reference object defines the moulded_surface for a Plate_design_definition.

4.3.112 Plate_design_definition to Panel_system_boundary

Each Plate_design_definition has border defined by one up to many Panel_system_boundary objects. A set of Panel_system_boundary objects defines the border for a Plate_design_definition.

4.3.113 Plate_design_definition to Plate

Each Plate_design_definition has defined_for defined by one up to many Plate objects. A set of Plate objects defines the defined_for for a Plate_design_definition.

4.3.114 Plate_design_definition to Plate_boundary_relationship

Each Plate_design_definition has border defined by one up to many Plate_boundary_relationship objects. A set of Plate_boundary_relationship objects defines the border for a Plate_design_definition.

4.3.115 Plate_design_definition to Surface

Each Plate_design_definition has moulded_surface defined by exactly one Surface object. A Surface object defines the moulded_surface for a Plate_design_definition.

4.3.116 Plate_functional_definition to Plate

Each Plate_functional_definition has defined_for defined by one up to many Plate objects. A set of Plate objects defines the defined_for for a Plate_functional_definition.

4.3.117 Plate_manufacturing_definition to Curve

Each Plate_manufacturing_definition has outer_contour defined by one up to many Curve objects. A set of Curve objects defines the outer_contour for a Plate_manufacturing_definition.

4.3.118 Plate_rejection to Plate_design_definition

Each Plate_rejection has subject defined by exactly one Plate_design_definition. A Plate_design_definition defines the subject for a Plate_rejection.

4.3.119 Plate_relationship to Plate

Each Plate_relationship has item_1 defined by exactly one Plate. A Plate defines the item_1 for a Plate_relationship.

4.3.120 Plate_renewal_definition to Plate

Each Plate_renewal_definition has defined_for defined by one up to many Plate objects. A set of Plate objects defines the defined_for for a Plate_renewal_definition.

4.3.121 Plate_strike_functional_definition to Plate_strike

Each Plate_strike_functional_definition has defined_for defined by one up to many Plate_strike objects. A set of Plate_strike objects defines the defined_for for a Plate_strike_functional_definition.

4.3.122 Position_feature_design_definition to Position_feature

Each Position_feature_design_definition has defined_for defined by one up to many Position_feature objects. A set of Position_feature objects defines the defined_for for a Position_feature_design_definition.

4.3.123 Position_feature_design_definition to Shape_representation

Each Position_feature_design_definition has representations defined by zero up to many Shape_representation objects. A set of Shape_representation objects defines the representations for a Position_feature_design_definition.

4.3.124 Position_feature_relationship to Position_feature

Each Position_feature_relationship has item_1 defined by exactly one Position_feature. A Position_feature defines the item_1 for a Position_feature_relationship.

4.3.125 Profile_approval to Profile_design_definition

Each Profile_approval has subject defined by exactly one Profile_design_definition. A Profile_design_definition defines the subject for a Profile_approval.

4.3.126 Profile_cross_section to Section_properties

Each Profile_cross_section has section_properties defined by zero or one Section_properties. A Section_properties defines the section_properties for a Profile_cross_section.

4.3.127 Profile_curve_trace_line to Curve

Each Profile_curve_trace_line has curve defined by exactly one Curve. A Curve defines the curve for a Profile_curve_trace_line.

4.3.128 Profile_design_definition to Axis2_placement_3d

Each Profile_design_definition has cross_section_placement defined by exactly one Axis2_placement_3d. An Axis2_placement_3d defines the cross_section_placement for a Profile_design_definition.

4.3.129 Profile_design_definition to Cartesian_point

Each Profile_design_definition has border defined by one up to many Cartesian_point objects. A set of Cartesian_point objects defines the border for a Profile_design_definition.

4.3.130 Profile_design_definition to External_instance_reference

Each Profile_design_definition has trace_line defined by exactly one External_instance_reference object. A External_instance_reference object defines the trace_line for a Profile_design_definition.

4.3.131 Profile_design_definition to Line

Each Profile_design_definition has border defined by one up to many Line objects. A set of Line objects defines the border for a Profile_design_definition.

4.3.132 Profile_design_definition to Point_on_curve

Each Profile_design_definition has border defined by one up to many Point_on_curve objects. A set of Point_on_curve objects defines the border for a Profile_design_definition.

4.3.133 Profile_design_definition to Profile

Each Profile_design_definition has defined_for defined by one up to many Profile objects. A set of Profile objects defines the defined_for for a Profile_design_definition.

4.3.134 Profile_design_definition to Profile_boundary_relationship

Each Profile_design_definition has border defined by one up to many Profile_boundary_relationship objects. A set of Profile_boundary_relationship objects defines the border for a Profile_design_definition.

4.3.135 Profile_design_definition to Profile_cross_section

Each Profile_design_definition has cross_section defined by exactly one Profile_cross_section. A Profile_cross_section defines the cross_section for a Profile_design_definition.

4.3.136 Profile_design_definition to Profile_curve_trace_line

Each Profile_design_definition has trace_line defined by exactly one Profile_curve_trace_line. A Profile_curve_trace_line defines the trace_line for a Profile_design_definition.

4.3.137 Profile_design_definition to Profile_trace_line_relationship

Each Profile_design_definition has trace_line defined by exactly one Profile_trace_line_relationship. A Profile_trace_line_relationship defines the trace_line for a Profile_design_definition.

4.3.138 Profile_design_definition to Twist_location

Each Profile_design_definition has twist defined by zero up to many Twist_location objects. A set of Twist_location objects defines the twist for a Profile_design_definition.

4.3.139 Profile_functional_definition to Profile

Each Profile_functional_definition has defined_for defined by one up to many Profile objects. A set of Profile objects defines the defined_for for a Profile_functional_definition.

4.3.140 Profile_manufacturing_definition to Curve

Each Profile_manufacturing_definition has inverse_bend_trace defined by zero up to many Curve objects. A set of Curve objects defines the inverse_bend_trace for a Profile_manufacturing_definition.

Each Profile_manufacturing_definition has outer_flange_contour defined by zero up to many Curve objects. A set of Curve objects defines the outer_flange_contour for a Profile_manufacturing_definition.

Each Profile_manufacturing_definition has outer_web_contour defined by zero up to many Curve objects. A set of Curve objects defines the outer_web_contour for a Profile_manufacturing_definition.

4.3.141 Profile_manufacturing_definition to Profile

Each Profile_manufacturing_definition has defined_for defined by one up to many Profile objects. A set of Profile objects defines the defined_for for a Profile_manufacturing_definition.

4.3.142 Profile_rejection to Profile_design_definition

Each Profile_rejection has subject defined by exactly one Profile_design_definition. A Profile_design_definition defines the subject for a Profile_rejection.

4.3.143 Profile_relationship to Profile

Each Profile_relationship has item_1 defined by exactly one Profile. A Profile defines the item_1 for a Profile_relationship.

4.3.144 Profile_renewal_definition to Profile

Each Profile_renewal_definition has defined_for defined by one up to many Profile objects. A set of Profile objects defines the defined_for for a Profile_renewal_definition.

4.3.145 Profile_trace_line_relationship to Line

Each Profile_trace_line_relationship has auxiliary_line defined by zero or one Line. A Line defines the auxiliary_line for a Profile_trace_line_relationship.

4.3.146 Property_bsu to Class_bsu

Each Property_bsu has name_scope defined by exactly one Class_bsu. A Class_bsu defines the name_scope for a Property_bsu.

4.3.147 Property_value to Property_bsu

Each Property_value has property_identifier defined by exactly one Property_bsu. A Property_bsu defines the property_identifier for a Property_value.

4.3.148 Reason_for_decision to Document_reference

Each Reason_for_decision has rule_reference defined by exactly one Document_reference. A Document_reference defines the rule_reference for a Reason_for_decision.

4.3.149 Regulation to Document_reference

Each Regulation has international_regulations defined by zero up to many Document_reference objects. A set of Document_reference objects defines the international_regulations for a Regulation.

Each Regulation has national_regulations defined by zero up to many Document_reference objects. A set of Document_reference objects defines the national_regulations for a Regulation.

Each Regulation has standards defined by zero up to many Document_reference objects. A set of Document_reference objects defines the standards for a Regulation.

4.3.150 Revision to Versionable_object

Each Revision has members defined by one up to many Versionable_object objects. A set of Versionable_object objects defines the members for a Revision.

4.3.151 Revision_with_context to Definable_object

Each Revision_with_context has context_of_revision defined by exactly one Definable_object. A Definable_object defines the context_of_revision for a Revision_with_context.

4.3.152 Seam to Panel_system

Each Seam has parent defined by exactly one Panel_system. A Panel_system defines the parent for a Seam.

4.3.153 Seam_design_definition to Seam

Each Seam_design_definition has border defined by zero up to two Seam objects. A list of Seam objects defines the border for a Seam_design_definition.

Each Seam_design_definition has defined_for defined by one up to many Seam objects. A set of Seam objects defines the defined_for for a Seam_design_definition.

4.3.154 Ship_material_property to Document_reference_with_address

Each Ship_material_property has material_reference defined by zero or one Document_reference_with_address. A Document_reference_with_address defines the material_reference for a Ship_material_property.

4.3.155 Ship_material_property to Item

Each Ship_material_property has defined_for defined by one up to many Item objects. A set of Item objects defines the defined_for for a Ship_material_property.

4.3.156 Shiptype to Ship

Each Shiptype has defined_for defined by one up to many Ship objects. A set of Ship objects defines the defined_for for a Shiptype.

4.3.157 Shipyard_designation to Organisation

Each Shipyard_designation has shipyard defined by exactly one Organisation. An Organisation defines the shipyard for a Shipyard_designation.

4.3.158 Si_unit to Dimensional_exponents

Each Si_unit has dimensions defined by exactly one Dimensional_exponents. A Dimensional_exponents defines the dimensions for a Si_unit.

4.3.159 Spacing_position_with_offset to Spacing_position

Each Spacing_position_with_offset has relating_spacing_position defined by exactly one Spacing_position. A Spacing_position defines the relating_spacing_position for a Spacing_position_with_offset.

4.3.160 Spacing_table to Spacing_position

Each Spacing_table has spacing_table_representations defined by zero up to many Spacing_position objects. A list of Spacing_position objects defines the spacing_table_representations for a Spacing_table.

4.3.161 Structural_added_material_boundary_relationship to Structural_added_material_feature

Each Structural_added_material_boundary_relationship has item_1 defined by exactly one Structural_added_material_feature. A Structural_added_material_feature defines the item_1 for a Structural_added_material_boundary_relationship.

4.3.162 Structural_added_material_feature_design_definition to Structural_added_material_feature

Each Structural_added_material_feature_design_definition has defined_for defined by one up to many Structural_added_material_feature objects. A set of Structural_added_material_feature objects defines the defined_for for a Structural_added_material_feature_design_definition.

4.3.163 Structural_class_approval_result to Reason_for_decision

Each Structural_class_approval_result has explanations defined by zero up to many Reason_for_decision objects. A set of Reason_for_decision objects defines the explanations for a Structural_class_approval_result.

4.3.164 Structural_class_conditional_approval to Reason_for_decision

Each Structural_class_conditional_approval has explanations defined by one up to many Reason_for_decision objects. A set of Reason_for_decision objects defines the explanations for a Structural_class_conditional_approval.

4.3.165 Structural_class_rejection to Definition

Each Structural_class_rejection has proposed_alternative defined by zero up to many Definition objects. A set of Definition objects defines the proposed_alternative for a Structural_class_rejection.

4.3.166 Structural_class_rejection to Reason_for_decision

Each Structural_class_rejection has explanations defined by one up to many Reason_for_decision objects. A set of Reason_for_decision objects defines the explanations for a Structural_class_rejection.

4.3.167 Structural_cutout_boundary_relationship to Structural_cutout

Each Structural_cutout_boundary_relationship has item_1 defined by exactly one Structural_cutout. A Structural_cutout defines the item_1 for a Structural_cutout_boundary_relationship.

4.3.168 Structural_feature to Item

Each Structural_feature has parent defined by exactly one Item. An Item defines the parent for a Structural_feature.

4.3.169 Structural_feature_relationship to Structural_feature

Each Structural_feature_relationship has item_1 defined by exactly one Structural_feature. A Structural_feature defines the item_1 for a Structural_feature_relationship.

4.3.170 Structural_manufacturing_feature to Structural_part

Each Structural_manufacturing_feature has parent defined by exactly one Structural_part. A Structural_part defines the parent for a Structural_manufacturing_feature.

4.3.171 Structural_part_design_definition to Shape_representation

Each Structural_part_design_definition has representations defined by zero up to many Shape_representation objects. A set of Shape_representation objects defines the representations for a Structural_part_design_definition.

4.3.172 Structural_part_design_definition to Structural_part

Each Structural_part_design_definition has defined_for defined by one up to many Structural_part objects. A set of Structural_part objects defines the defined_for for a Structural_part_design_definition.

4.3.173 Structural_part_functional_definition to Structural_part

Each Structural_part_functional_definition has defined_for defined by one up to many Structural_part objects. A set of Structural_part objects defines the defined_for for a Structural_part_functional_definition.

4.3.174 Structural_part_joint to Structural_part_connection_implementation

Each Structural_part_joint has realization_of defined by exactly one Structural_part_connection_implementation. Structural_part_connection_implementation defines the realization for a Structural_part_joint.

4.3.175 Structural_part_joint_design_definition to Structural_part_joint

Each Structural_part_joint_design_definition has defined_for defined by one up to many Structural_part_joint objects. A set of Structural_part_joint objects defines the defined_for for a Structural_part_joint_design_definition.

4.3.176 Structural_part_manufacturing_definition to Curve

Each Structural_part_manufacturing_definition has annotation defined by zero up to many Curve objects. A set of Curve objects defines the annotation for a Structural_part_manufacturing_definition.

Each Structural_part_manufacturing_definition has inner_contours defined by zero up to many Curve objects. A set of Curve objects defines the inner_contours for a Structural_part_manufacturing_definition.

Each Structural_part_manufacturing_definition has layout_marks defined by zero up to many Curve objects. A set of Curve objects defines the layout_marks for a Structural_part_manufacturing_definition.

4.3.177 Structural_part_manufacturing_definition to Structural_part

Each Structural_part_manufacturing_definition has defined_for defined by one up to many Structural_part objects. A set of Structural_part objects defines the defined_for for a Structural_part_manufacturing_definition.

4.3.178 Structural_part_penetration_relationship to Structural_cutout

Each Structural_part_penetration_relationship has penetration_result defined by one up to many Structural_cutout objects. A set of Structural_cutout objects defines the penetration_result for a Structural_part_penetration_relationship.

4.3.179 Structural_part_penetration_relationship to Structural_part

Each Structural_part_penetration_relationship has item_1 defined by exactly one Structural_part. A Structural_part defines the item_1 for a Structural_part_penetration_relationship.

4.3.180 Structural_part_relationship to Structural_part

Each Structural_part_relationship has item_1 defined by exactly one Structural_part. A Structural_part defines the item_1 for a Structural_part_relationship.

4.3.181 Structural_part_symmetry_relationship to Plane

Each Structural_part_symmetry_relationship has mirroring_plane defined by exactly one Plane. A Plane defines the mirroring_plane for a Structural_part_symmetry_relationship.

4.3.182 Structural_part_symmetry_relationship to Structural_part

Each Structural_part_symmetry_relationship has item_2 defined by exactly one Structural_part. A Structural_part defines the item_2 for a Structural_part_symmetry_relationship.

4.3.183 Structural_system_adjacency_relationship to Structural_system

Each Structural_system_adjacency_relationship has item_2 defined by exactly one Structural_system. A Structural_system defines the item_2 for a Structural_system_adjacency_relationship.

4.3.184 Structural_system_design_definition to Shape_representation

Each Structural_system_design_definition has representations defined by zero up to many Shape_representation objects. A set of Shape_representation objects defines the representations for a Structural_system_design_definition.

4.3.185 Structural_system_design_definition to Structural_system

Each Structural_system_design_definition has defined_for defined by one up to many Structural_system objects. A set of Structural_system objects defines the defined_for for a Structural_system_design_definition.

4.3.186 Structural_system_functional_definition to Structural_system

Each Structural_system_functional_definition has defined_for defined by one up to many Structural_system objects. A set of Structural_system objects defines the defined_for for a Structural_system_functional_definition.

4.3.187 Structural_system_penetration_relationship to Structural_cutout

Each Structural_system_penetration_relationship has penetration_result defined by one up to many Structural_cutout objects. A set of Structural_cutout objects defines the penetration_result for a Structural_system_penetration_relationship.

4.3.188 Structural_system_penetration_relationship to Structural_system

Each Structural_system_penetration_relationship has item_1 defined by exactly one Structural_system. A Structural_system defines the item_1 for a Structural_system_penetration_relationship.

4.3.189 Structural_system_relationship to Structural_system

Each Structural_system_relationship has item_1 defined by exactly one Structural_system. A Structural_system defines the item_1 for a Structural_system_relationship.

4.3.190 Structural_system_symmetry_relationship to Plane

Each Structural_system_symmetry_relationship has mirroring_plane defined by exactly one Plane. A Plane defines the mirroring_plane for a Structural_system_symmetry_relationship.

4.3.191 Structural_system_symmetry_relationship to Structural_system

Each Structural_system_symmetry_relationship has item_2 defined by exactly one Structural_system. A Structural_system defines the item_2 for a Structural_system_symmetry_relationship.

4.3.192 Structural_weld_shrinkage_allowance_feature_design_definition to Structural_weld_shrinkage_allowance_feature

Each Structural_weld_shrinkage_allowance_feature_design_definition has defined_for defined by one up to many Structural_weld_shrinkage_allowance_feature objects. A set of Structural_weld_shrinkage_allowance_feature objects defines the defined_for for a Structural_weld_shrinkage_allowance_feature_design_definition.

4.3.193 System_design_definition to System

Each System_design_definition has defined_for defined by one up to many System objects. A set of System objects defines the defined_for for a System_design_definition.

4.3.194 System_relationship to System

Each System_relationship has item_1 defined by exactly one System. A System defines the item_1 for a System_relationship.

4.3.195 Transversal_table to Transversal_position

Each Transversal_table has spacing_table_representations defined by zero up to many Transversal_position objects. A list of Transversal_position objects defines the spacing_table_representations for a Transversal_table.

4.3.196 Twist_location to Direction

Each Twist_location has orientation defined by exactly one Direction. A Direction defines the orientation for a Twist_location.

4.3.197 Twist_location to Point_on_curve

Each Twist_location has location defined by exactly one Point_on_curve. A Point_on_curve defines the location for a Twist_location.

4.3.198 Version_creation to Versionable_object

Each Version_creation has base defined by zero up to many Versionable_object objects. A set of Versionable_object objects defines the base for a Version_creation.

Each Version_creation has subject defined by exactly one Versionable_object. A Versionable_object defines the subject for a Version_creation.

4.3.199 Version_deletion to Versionable_object

Each Version_deletion has subject defined by exactly one Versionable_object. A Versionable_object defines the subject for a Version_deletion.

4.3.200 Version_history to Version_relationship

Each Version_history has relationships defined by zero up to many Version_relationship objects. A set of Version_relationship objects defines the relationships for a Version_history.

4.3.201 Version_history to Versionable_object

Each Version_history has current_version defined by exactly one Versionable_object. A Versionable_object defines the current_version for a Version_history.

Each Version_history has versions defined by one up to many Versionable_object objects. A set of Versionable_object objects defines the versions for a Version_history.

4.3.202 Version_modification to Versionable_object

Each Version_modification has base defined by one up to many Versionable_object objects. A set of Versionable_object objects defines the base for a Version_modification.

Each Version_modification has subject defined by exactly one Versionable_object. A Versionable_object defines the subject for a Version_modification.

4.3.203 Version_relationship to Versionable_object

Each Version_relationship has predecessor defined by exactly one Versionable_object. A Versionable_object defines the predecessor for a Version_relationship.

Each Version_relationship has successor defined by exactly one Versionable_object. A Versionable_object defines the successor for a Version_relationship.

4.3.204 Vertical_table to Vertical_position

Each Vertical_table has spacing_table_representations defined by zero up to many Vertical_position objects. A list of Vertical_position objects defines the spacing_table_representations for a Vertical_table.

4.3.205 Weight_and_centre_of_gravity to Cartesian_point

Each Weight_and_centre_of_gravity has centre_of_gravity defined by exactly one Cartesian_point. A Cartesian_point defines the centre_of_gravity for a Weight_and_centre_of_gravity.

4.3.206 Weight_and_centre_of_gravity to Moment_3d

Each Weight_and_centre_of_gravity has moment defined by exactly one Moment_3d. A Moment_3d defines the moment for a Weight_and_centre_of_gravity.

4.3.207 Weld to Welded_joint

Each Weld has realization_of defined by exactly one Welded_joint. A Welded_joint defines the realization_of for a Weld.

4.3.208 Weld_design_definition to Curve

Each Weld_design_definition has weld_geometry defined by exactly one Curve. A Curve defines the weld_geometry for a Weld_design_definition.

4.3.209 Weld_design_definition to Weld

Each Weld_design_definition has defined_for defined by one up to many Weld objects. A set of Weld objects defines the defined_for for a Weld_design_definition.

4.3.210 Weld_filler_material to Electrode_chemical_composition

Each Weld_filler_material has chemical_composition defined by zero or one Electrode_chemical_composition. An Electrode_chemical_composition defines the chemical_composition for a Weld_filler_material.

filler_material.

4.3.211 Weld_filler_material to Weld

Each Weld_filler_material has defined_for defined by one up to many Weld objects. A set of Weld objects defines the defined_for for a Weld_filler_material.

4.3.212 Weld_manufacturing_definition to Document

Each Weld_manufacturing_definition has welding_procedures defined by one up to many Document objects. A set of Document objects defines the welding_procedures for a Weld_manufacturing_definition.

4.3.213 Weld_manufacturing_definition to Vector

Each Weld_manufacturing_definition has torch_vector defined by exactly one Vector. A Vector defines the torch_vector for a Weld_manufacturing_definition.

4.3.214 Weld_manufacturing_definition to Weld

Each Weld_manufacturing_definition has defined_for defined by one up to many Weld objects. A set of Weld objects defines the defined_for for a Weld_manufacturing_definition.

4.3.215 Weld_manufacturing_definition to Weld_testing

Each Weld_manufacturing_definition has weld_test defined by one up to many Weld_testing objects. A set of Weld_testing objects defines the weld_test for a Weld_manufacturing_definition.

4.3.216 Weld_manufacturing_definition to Welding_sequence

Each Weld_manufacturing_definition has sequences defined by one up to many Welding_sequence objects. A set of Welding_sequence objects defines the sequences for a Weld_manufacturing_definition.

4.3.217 Weld_testing to Document

Each Weld_testing has test_sequence defined by zero up to many Document objects. A set of Document objects defines the test_sequence for a Weld_testing.

4.3.218 Welded_joint_design_definition to Shape_representation

Each Welded_joint_design_definition has representations defined by zero up to many Shape_representation objects. A set of Shape_representation objects defines the representations for a Welded_joint_design_definition.

4.3.219 Welded_joint_design_definition to Welded_joint

Each Welded_joint_design_definition has defined_for defined by one up to many Welded_joint objects. A set of Welded_joint objects defines the defined_for for a Welded_joint_design_definition.

5 Application interpreted model

5.1 Mapping table

This clause contains the mapping table that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM constructs (see annex A). The mapping table is organized in five columns.

Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lower case.

Column 2) AIM element: Name of an AIM element as it appears in the AIM (see annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name> . <attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of an application assertion map to the same AIM element. The term 'PATH' indicates that the application assertion maps to the entire reference path.

Column 3) Source: For those AIM elements that are interpreted from the integrated resources or the application interpreted constructs, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part. Entities or types that are defined within the integrated resources have an AIC as the source reference if the use of the entity or type for the mapping is within the scope of the AIC.

Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element or reference path. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. The expanded names of the rules are listed after the table.

Column 5) Reference path: To describe fully the mapping of an application object, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path up to its supertype from an integrated resource is specified.

For the expression of reference paths the following notational conventions apply:

- a) [] : enclosed section constrains multiple AIM elements or sections of the reference path are required to satisfy an information requirement;

- b) `()` : enclosed section constrains multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;
- c) `{}` : enclosed section constrains the reference path to satisfy an information requirement;
- d) `<>` : enclosed section constrains at one or more required reference path;
- e) `||` : enclosed section constrains the supertype entity;
- f) `->` : attribute references the entity or select type given in the following row;
- g) `<-` : entity or select type is referenced by the attribute in the following row;
- h) `[i]` : attribute is an aggregation of which a single member is given in the following row;
- i) `[n]` : attribute is an aggregation of which member `n` is given in the following row;
- j) `=>` : entity is a supertype of the entity given in the following row;
- k) `<=` : entity is a subtype of the entity given in the following row;
- l) `=` : the string, select, or enumeration type is constrained to a choice or value;
- m) `\` : the reference path expression continues on the next line.

Table 1 - Mapping table for class_approvals UoF

Application Element	AIM element	Source	Rules	Reference Path
CORRUGATED_PART_APPROVAL	group	41	1	{[group group.name = '.UNUSED.']} {[CLASS<group, 'corrugated part approval', 'structural class approval']>] [CLASS<group, 'structural class approval', 'structural class approval result']>] [CLASS<group, 'structural class approval result', 'approval history']>] [ROOT_CLASS<group, 'approval history']>]}
corrugated_part_approval to corrugated_part_design_definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'corrugated part design definition'>}
- from STRUCTURAL_CLASS_APPROVAL				
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
corrugated_part_approval to reason_for_decision (as explanations) (SET [0:?])	<i>see supertype</i>			
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
corrugated_part_approval to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
CORRUGATED_PART_REJECTION	group	41	1	{[CLASS<group, 'corrugated part rejection', 'structural class rejection']>] [CLASS<group, 'structural class rejection', 'structural class approval result']>] [CLASS<group, 'structural class approval result', 'approval history']>] [ROOT_CLASS<group, 'approval history']>]}

Application Element	AIM element	Source	Rules	Reference Path
				[group.name = '.UNUSED.']}
corrugated_part_rejection to corrugated_part_design_definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'corrugated part design definition'>}
- from STRUCTURAL_CLASS_REJECTION				
corrugated_part_rejection to definition (as proposed_alternative) (SET [0:?])	PATH		5, 10	group <- GROUPS<approval, 'approvals'> approval <- (APPROVES<extended_product_definition_shape, 'proposed alternative'> {CLASS_ID<extended_product_definition_shape, 'corrugated part design definition'>})
corrugated_part_rejection to reason_for_decision (as explanations) (SET [1:?])	<i>see supertype</i>			
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
corrugated_part_rejection to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
HULL_CROSS_SECTION_APPROVAL	[group] [representation]	41, 43	1, 4	{representation {[representation.id = '.UNUSED.' [representation.name = '.UNUSED.']} LINK_TO_GROUP<representation> {group.name = '.UNUSED.'} [CLASS<group, 'hull cross section approval', 'hull cross section result'>] [CLASS<group, 'hull cross section result', 'structural class approval result'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<group, 'structural class approval result', 'approval history'>] [ROOT_CLASS<group, 'approval history'>]]
- from HULL_CROSS_SECTION_RESULT				
actual_first_moment_horizontal	<i>see supertype</i>			
actual_first_moment_vertical	<i>see supertype</i>			
actual_inertia_cross	<i>see supertype</i>			
actual_inertia_horizontal	<i>see supertype</i>			
actual_inertia_torsion	<i>see supertype</i>			
actual_inertia_vertical	<i>see supertype</i>			
actual_neutral_axis_position_horizontal	<i>see supertype</i>			
actual_neutral_axis_position_vertical	<i>see supertype</i>			
actual_section_area	<i>see supertype</i>			
actual_section_modulus_bottom	<i>see supertype</i>			
actual_section_modulus_coaming	<i>see supertype</i>			
actual_section_modulus_deck	<i>see supertype</i>			
actual_stress_bottom	<i>see supertype</i>			
actual_stress_coaming	<i>see supertype</i>			
actual_stress_deck	<i>see supertype</i>			
actual_twist_centre_horizontal	<i>see supertype</i>			
actual_twist_centre_vertical	<i>see supertype</i>			
direction_largest_inertia	<i>see supertype</i>			
direction_least_inertia	<i>see supertype</i>			
max_moment	<i>see supertype</i>			
max_shear_force	<i>see supertype</i>			
meter_weight	<i>see supertype</i>			
min_moment	<i>see supertype</i>			
min_shear_force	<i>see supertype</i>			
required_section_modulus_bottom	<i>see supertype</i>			
required_section_modulus_coaming	<i>see supertype</i>			
required_section_modulus_deck	<i>see supertype</i>			
required_stress_bottom	<i>see supertype</i>			
required_stress_coaming	<i>see supertype</i>			
required_stress_deck	<i>see supertype</i>			
hull_cross_section_approval_to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
hull_cross_section_design_definition (as subject)				
- from STRUCTURAL_CLASS_APPROVAL				
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
hull_cross_section_approval to reason_for_decision (as explanations) (SET [0:?])	<i>see supertype</i>			
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
hull_cross_section_approval to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
HULL_CROSS_SECTION_REJECTION	[group] [representation]	41, 43	1, 4	{representation {[representation.id = 'UNUSED.']} [representation.name = 'UNUSED.']} LINK_TO_GROUP<representation> {group.name = 'UNUSED.'} [CLASS<group, 'hull cross section rejection', 'hull cross section result'>] [CLASS<group, 'hull cross section result', 'structural class approval result'>] [CLASS<group, 'structural class approval result', 'approval history'>] [ROOT_CLASS<group, 'approval history'>]]
- from HULL_CROSS_SECTION_RESULT				
actual_first_moment_horizontal	<i>see supertype</i>			
actual_first_moment_vertical	<i>see supertype</i>			
actual_inertia_cross	<i>see supertype</i>			
actual_inertia_horizontal	<i>see supertype</i>			
actual_inertia_torsion	<i>see supertype</i>			
actual_inertia_vertical	<i>see supertype</i>			
actual_neutral_axis_position_horizontal	<i>see supertype</i>			
actual_neutral_axis_position_vertical	<i>see supertype</i>			
actual_section_area	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
actual_section_modulus_bottom	<i>see supertype</i>			
actual_section_modulus_coaming	<i>see supertype</i>			
actual_section_modulus_deck	<i>see supertype</i>			
actual_stress_bottom	<i>see supertype</i>			
actual_stress_coaming	<i>see supertype</i>			
actual_stress_deck	<i>see supertype</i>			
actual_twist_centre_horizontal	<i>see supertype</i>			
actual_twist_centre_vertical	<i>see supertype</i>			
direction_largest_inertia	<i>see supertype</i>			
direction_least_inertia	<i>see supertype</i>			
max_moment	<i>see supertype</i>			
max_shear_force	<i>see supertype</i>			
meter_weight	<i>see supertype</i>			
min_moment	<i>see supertype</i>			
min_shear_force	<i>see supertype</i>			
required_section_modulus_bottom	<i>see supertype</i>			
required_section_modulus_coaming	<i>see supertype</i>			
required_section_modulus_deck	<i>see supertype</i>			
required_stress_bottom	<i>see supertype</i>			
required_stress_coaming	<i>see supertype</i>			
required_stress_deck	<i>see supertype</i>			
hull_cross_section_rejection to hull_cross_section_design_definition (as subject)	<i>see supertype</i>			
- from STRUCTURAL_CLASS_REJECTION				
hull_cross_section_rejection to definition (as proposed_alternative) (SET [0:?])	PATH		5, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'proposed alternative'> {CLASS_ID<extended_product_definition_shape, 'hull cross section design definition'>}
hull_cross_section_rejection to reason_for_decision (as explanations) (SET [1:?])	<i>see supertype</i>			
- from STRUCTURAL_CLASS_APPROVAL_RESULT				

Application Element	AIM element	Source	Rules	Reference Path
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
hull_cross_section_rejection to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
HULL_CROSS_SECTION_RESULT	[group] [representation]	41, 43	1, 4, 9	{representation {[representation.id = '.UNUSED.'] [representation.name = '.UNUSED.']} LINK_TO_GROUP<representation> {group.name = '.UNUSED.'} [CLASS<group, 'hull cross section result', 'structural class approval result']> [CLASS<group, 'structural class approval result', 'approval history']> [ROOT_CLASS<group, 'approval history']>]}
actual_first_moment_horizontal	value_representation_item.valu e_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual first moment horizontal', 'moment unit'>
actual_first_moment_vertical	value_representation_item.valu e_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual first moment vertical', 'moment unit'>
actual_inertia_cross	value_representation_item.valu e_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual inertia cross', 'moment unit'>
actual_inertia_horizontal	value_representation_item.valu e_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual inertia horizontal', 'inertia moment unit'>
actual_inertia_torsion	value_representation_item.valu e_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual inertia torsion', 'inertia moment unit'>
actual_inertia_vertical	value_representation_item.valu	43		representation

Application Element	AIM element	Source	Rules	Reference Path
	e_component			representation.items [i] -> VAL_REP_ITEM_CD<'actual inertia vertical', 'inertia moment unit'>
actual_neutral_axis_position_horizontal	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'actual neutral axis position horizontal', length_measure>
actual_neutral_axis_position_vertical	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'actual neutral axis position vertical', length_measure>
actual_section_area	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'actual neutral axis position horizontal', area_measure>
actual_section_modulus_bottom	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual section modulus bottom', 'section modulus unit'>
actual_section_modulus_coaming	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual section modulus coaming', 'section modulus unit'>
actual_section_modulus_deck	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual section modulus deck', 'section modulus unit'>
actual_stress_bottom	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual stress bottom', 'stress unit'>
actual_stress_coaming	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual stress coaming', 'stress unit'>
actual_stress_deck	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'actual stress deck', 'stress unit'>

Application Element	AIM element	Source	Rules	Reference Path
actual_twist_centre_horizontal	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'actual twist centre horizontal', length_measure>
actual_twist_centre_vertical	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'actual twist centre vertical', length_measure>
direction_largest_inertia	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'direction largest inertia', plane_angle_measure>
direction_least_inertia	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'direction least inertia', plane_angle_measure>
max_moment	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'max moment', 'moment unit'>
max_shear_force	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'max shear force', 'force unit'>
meter_weight	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'meter weight', mass_measure>
min_moment	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'min moment', 'moment unit'>
min_shear_force	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'min shear force', 'force unit'>
required_section_modulus_bottom	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'required section modulus bottom', 'section modulus unit'>
required_section_modulus_coaming	value_representation_item.value_component	43		representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				VAL_REP_ITEM_CD<'required section modulus coaming', 'section modulus unit'>
required_section_modulus_deck	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'required section modulus deck', 'section modulus unit'>
required_stress_bottom	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'required stress bottom', 'stress unit'>
required_stress_coaming	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'required stress coaming', 'stress unit'>
required_stress_deck	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM_CD<'required stress deck', 'stress unit'>
hull_cross_section_result to hull_cross_section_design_definition (as subject)	PATH		2, 3	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'hull cross section design definition'>}
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
hull_cross_section_result to reason_for_decision (as explanations) (SET [0:?])	<i>see supertype</i>			
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
hull_cross_section_result to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
PLATE_APPROVAL	group	41	1	{[CLASS<group, 'plate approval', 'structural class approval'>] [CLASS<group, 'structural class approval', 'structural class approval result'>] [CLASS<group, 'structural class approval result', 'approval

Application Element	AIM element	Source	Rules	Reference Path
				history'>] [ROOT_CLASS<group, 'approval history'>] [group.name = '.UNUSED.']]}
plate_approval to plate_design_definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'plate design definition'>}
- from STRUCTURAL_CLASS_APPROVAL				
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
plate_approval to reason_for_decision (as explanations) (SET [0:?])	<i>see supertype</i>			
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
plate_approval to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
PLATE_REJECTION	group	41	1	{[CLASS<group, 'plate rejection', 'structural class rejection'>] [CLASS<group, 'structural class rejection', 'structural class approval result'>] [CLASS<group, 'structural class approval result', 'approval history'>] [ROOT_CLASS<group, 'approval history'>] [group.name = '.UNUSED.']]}
plate_rejection to plate_design_definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES< extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'plate design definition'>}
- from STRUCTURAL_CLASS_REJECTION				
plate_rejection to definition (as proposed_alternative) (SET [0:?])	PATH		5, 10	group <- GROUPS<approval, 'approvals'>

Application Element	AIM element	Source	Rules	Reference Path
				approval <- (APPROVES<extended_product_definition_shape, 'proposed alternative'> {CLASS_ID<extended_product_definition_shape, 'plate design definition'>})
plate_rejection to reason_for_decision (as explanations) (SET [1:?])	<i>see supertype</i>			
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
plate_rejection to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
PLATE_RENEWAL_DEFINITION	extended_product_definition_shape	218	4, 7	{extended_product_definition_shape <= product_definition_shape} {[CLASS<extended_product_definition_shape, 'plate renewal definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
renewal_thickness	value_representation_item.value_component	43		extended_product_definition_shape <= product_definition_shape <= property_definition represented_definition = property_definition property_definition <- PDR<'plate renewal definition attributes'> -> representation representation.items [i] -> VAL_REP_ITEM<'renewal thickness', positive_length_measure>
rule_thickness	value_representation_item.value_component	43		extended_product_definition_shape <= product_definition_shape <= property_definition represented_definition = property_definition property_definition <-

Application Element	AIM element	Source	Rules	Reference Path
				PDR<'plate renewal definition attributes'> -> representation representation.items [i] -> VAL_REP_ITEM<'rule thickness', positive_length_measure>
plate_renewal_definition to plate (as defined_for) (SET [1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'plate'>}
- from DEFINITION				
plate_renewal_definition to global_id (as id)	extended_product_definition_shape.id	218		
plate_renewal_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROFILE_APPROVAL	group	41	1	{[CLASS<group, 'profile approval', 'structural class approval'>] [CLASS<group, 'structural class approval', 'structural class approval result'>] [CLASS<group, 'structural class approval result', 'approval history'>] [ROOT_CLASS<group, 'approval history'>] [group.name = '.UNUSED.']}
profile_approval to profile_design_definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'profile design definition'>}
- from STRUCTURAL_CLASS_APPROVAL				
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
profile_approval to reason_for_decision (as	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
explanations) (SET [0:?])				
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
profile_approval to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
PROFILE_REJECTION	group	41	1	{[CLASS<group, 'profile rejection', 'structural class rejection'>] [CLASS<group, 'structural class rejection', 'structural class approval result'>] [CLASS<group, 'structural class approval result', 'approval history'>] [ROOT_CLASS<group, 'approval history'>] [group.name = 'UNUSED.']}
profile_rejection to profile_design_definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {CLASS_ID<extended_product_definition_shape, 'profile design definition'>}
- from STRUCTURAL_CLASS_REJECTION				
profile_rejection to definition (as proposed_alternative) (SET [0:?])	PATH		5, 10	group <- GROUPS<approval, 'approvals'> approval <- (APPROVES<extended_product_definition_shape, 'proposed alternative'> {CLASS_ID<extended_product_definition_shape, 'profile design definition'>}
profile_rejection to reason_for_decision (as explanations) (SET [1:?])	<i>see supertype</i>			
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
profile_rejection to approval_event (as approvals)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(LIST [1:?] OF UNIQUE)				
PROFILE_RENEWAL_DEFINITION	extended_product_definition_shape	218	4, 8	{extended_product_definition_shape <= product_definition_shape} {[CLASS<extended_product_definition_shape, 'profile renewal definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
renewal_modulus	value_representation_item.value_component	43		extended_product_definition_shape <= product_definition_shape <= property_definition represented_definition = property_definition property_definition <- PDR<'profile renewal definition attributes'> -> representation representation.items [i] -> VAL_REP_ITEM_CD<'renewal modulus', 'section modulus unit'>
rule_inertia	value_representation_item.value_component	43		extended_product_definition_shape <= product_definition_shape <= property_definition represented_definition = property_definition property_definition <- PDR<'profile renewal definition attributes'> -> representation representation.items [i] -> VAL_REP_ITEM_CD<'rule inertia', 'inertia moment unit'>
rule_section_modulus	value_representation_item.value_component	43		extended_product_definition_shape <= product_definition_shape <= property_definition represented_definition = property_definition property_definition <- PDR<'profile renewal definition attributes'> -> representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				VAL_REP_ITEM_CD<'rule section modulus', 'section modulus unit'>
profile_renewal_definition to profile (as defined_for)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition (product_definition {CLASS_ID<product_definition, 'profile'>}) (=> product_definition_with_associated_documents {CLASS_ID<product_definition_with_associated_documents, 'profile'>})
- from DEFINITION				
profile_renewal_definition to global_id (as id)	extended_product_definition_shape.id	218		
profile_renewal_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
REASON_FOR_DECISION	applied_document_reference	218		{applied_document_reference <= document_reference } {ROOT_CLASS<applied_document_reference, 'reason for decision'>}
annotations	document.description	41		applied_document_reference <= document_reference document_reference.assigned_document -> document document.description
reason_for_decision to document_reference (as rule_reference)	IDENTICAL MAPPING			
STRUCTURAL_CLASS_APPROVAL (ABS)	group	41		<i>maps OR subtypes</i>
- from				

Application Element	AIM element	Source	Rules	Reference Path
STRUCTURAL_CLASS_APPROVAL_RESULT				
structural_class_approval to reason_for_decision (as explanations) (SET [0:?])	<i>see supertype</i>			
- from APPROVAL_HISTORY				
status (DER)	approval_status	41		<pre> group <- GROUPS<approval, 'approvals'> approval approval.status -> approval_status (approval_status.name = 'approved') (approval_status.name = 'user defined')]] </pre>
structural_class_approval to definition (as subject)	<i>see subtypes</i>			
structural_class_approval to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
STRUCTURAL_CLASS_APPROVAL_RESULT (ABS) 1: structural_class_approval, structural_class_rejection, structural_class_conditional_approval 2: hull_cross_section_result	1: group 2: [group] [representation]	41		<i>maps OR subtypes</i>
structural_class_approval_result to reason_for_decision (as explanations) (SET [0:?])	PATH			<pre> group document_reference_item = group document_reference_item <- applied_document_reference.items[i] applied_document_reference { <= document_reference document_reference.role -> object_role [object_role.name = 'explanations'] [object_role.description = '!UNUSED. '] </pre>
- from APPROVAL_HISTORY				
status (DER)	<i>see subtypes</i>			

Application Element	AIM element	Source	Rules	Reference Path
structural_class_approval_result to definition (as subject)	<i>see subtypes</i>			
structural_class_approval_result to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
STRUCTURAL_CLASS_CONDITIONAL_APPROVAL	group	41	1	{[CLASS<group, 'structural class conditional approval', 'structural class approval result'>] [CLASS<group, 'structural class approval result', 'approval history'>] [ROOT_CLASS<group, 'approval history'>] [group.name = '.UNUSED.']}
structural_class_conditional_approval to reason_for_decision (as explanations) (SET [1:?])	<i>see supertype</i>			
- from APPROVAL_HISTORY				
status (DER)	<i>see supertype</i>			
structural_class_conditional_approval to definition (as subject)	PATH		2, 3, 10	group <- GROUPS<approval, 'approvals'> approval <- APPROVES<extended_product_definition_shape, 'subject'> {(CLASS_ID<extended_product_definition_shape, 'structural part design definition'>) (CLASS_ID<extended_product_definition_shape, 'structural system design definition'>) (CLASS_ID<extended_product_definition_shape, 'hull cross section design definition'>)}
structural_class_conditional_approval to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			
STRUCTURAL_CLASS_REJECTION (ABS)	group	41		<i>maps OR subtypes</i>
structural_class_rejection to definition (as proposed_alternative) (SET [0:?])	<i>see subtypes</i>		5	
structural_class_rejection to reason_for_decision (as	PATH		6	group

Application Element	AIM element	Source	Rules	Reference Path
explanations) (SET [1:?])				document_reference_item = group document_reference_item <- applied_document_reference.items[i] applied_document_reference { <= document_reference document_reference.role -> object_role [object_role.name = 'explanations'] [object_role.description = 'UNUSED. ']}
- from STRUCTURAL_CLASS_APPROVAL_RESULT				
- from APPROVAL_HISTORY				
status (DER)	approval_status	41		group <- GROUPS<approval, 'approvals'> approval approval.status -> approval_status (approval_status.name = 'rejected') (approval_status.name = 'user defined')}]}
structural_class_rejection to definition (as subject)	<i>see subtypes</i>			
structural_class_rejection to approval_event (as approvals) (LIST [1:?] OF UNIQUE)	<i>see supertype</i>			

Table 2 - Mapping table for UoF configuration_management

Application Element	AIM element	Source	Rules	Reference Path
APPROVAL_EVENT	approval	41		
result	approval_status.name	41		<pre> approval approval.status -> approval_status approval_status.name { (approval_status.name = 'noted') (approval_status.name = 'unapproved') (approval_status.name = 'conditionally_approved') (approval_status.name = 'approved') (approval_status.name = 'rejected')} </pre>
user_defined_result	approval_status.name	41		<pre> approval approval.status -> approval_status approval_status.name </pre>
approval_event to approval_history (as approval_reference) <INV>	PATH			<pre> approval group_item = approval group_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment { group_assignment.role -> object_role [object_role.name = 'approval'] [object_role.description = '.UNUSED.']} group_assignment.assigned_group -> group </pre>
- from EVENT				
description	approval.level	41		
approval_event to person_and_organisation (as caused_by)	PATH		17	<pre> approval <- approval_person_organization.authorised_approval approval_person_organization </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {approval_person_organization approval_person_organization.role -> approval_role approval_role.role='caused by'} approval_person_organization.person_organization -> person_organization_select = person_and_organization person_and_organization </pre>
approval_event to date_and_time (as caused_when)	PATH		18	<pre> approval <- approval_date_time.dated_approval approval_date_time approval_date_time.date_time -> date_time_select = date_and_time date_and_time </pre>
APPROVAL_HISTORY	group	41		{ROOT_CLASS<group, 'approval history'>}
approval_history to approval_event (as approvals) <L[1:?] UNIQUE>	PATH		11	<pre> group <- GROUPS<approval, 'approvals'> </pre>
approval_history to definition (as subject)	PATH		12, 14	<pre> group <- GROUPS<approval, 'approval'> <- (APPROVES<product_definition, 'subject'> {CLASS_ID<product_definition, 'definition' }) (APPROVES<property_definition, 'subject'> {CLASS_ID<property_definition, 'definition' }) (APPROVES<representation, 'subject'> {CLASS_ID<representation, 'definition' }) </pre>
REVISION	group	41		{[CLASS<group, 'revision', 'versionable object'>] [ROOT_CLASS<group, 'versionable object'>]}
name	group.name	41		
reason	group.description	41	15	
revision to versionable_object (as members) <SET[1:?]>	PATH		16	<pre> group <- (RELATE_GROUP_2_VO<representation, 'members'>) (RELATE_GROUP_2_VO<product_definition, 'members'>) (RELATE_GROUP_2_VO<property_definition, </pre>

Application Element	AIM element	Source	Rules	Reference Path
				'members'>) (RELATE_GROUP_2_VO<shape_aspect, 'members' >) (RELATE_GROUP_2_VO<product_category, 'members'>) (RELATE_GROUP_2_VO<document, 'members' >) (RELATE_GROUP_2_VO<product_definition_relationship, 'members' >) (RELATE_GROUP_2_VO<shape_aspect_relationship, 'members' >)
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.as signed_id	41		VERSION_ID< group >
REVISION_WITH_CONTEXT	group	41		{[CLASS<group, 'revision with context', 'revision'>] [CLASS<group, 'revision', 'versionable object'>] [ROOT_CLASS<group, 'versionable object'>]}
revision_with_context to definable_object (as context_of_revision)	PATH		13	group <- (RELATE_GROUP_2_DO< product, 'context of revision'>) (RELATE_GROUP_2_DO<product_definition, 'context of revision'>) (RELATE_GROUP_2_DO<product_definition_relationship, 'context of revision'>) (RELATE_GROUP_2_DO< shape_aspect, 'context of revision'>) (RELATE_GROUP_2_DO< shape_aspect_relationship, 'context of revision'>)
- from REVISION				
name	<i>see supertype</i>			
reason	<i>see supertype</i>			
revision to versionable_object (as members) <SET[1:?]>	PATH			<i>see supertype</i>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.as	41		VERSION_ID< group >

Application Element	AIM element	Source	Rules	Reference Path
	signed_id			
VERSION_HISTORY	group	41		{ROOT_CLASS<group, 'version history'>}
version_history to versionable_object (as current_version)	PATH		19	group <- (RELATE_GROUP_2_VO<representation, 'versions'>) (RELATE_GROUP_2_VO<product_definition, 'versions'>) (RELATE_GROUP_2_VO<property_definition, 'versions'>) (RELATE_GROUP_2_VO<shape_aspect, 'versions'>) (RELATE_GROUP_2_VO<product_category, 'versions', >) (RELATE_GROUP_2_VO<document, 'versions'>) (RELATE_GROUP_2_VO<product_definition_relationship, 'versions'>) (RELATE_GROUP_2_VO<shape_aspect_relationship, 'versions'>)
version_history to version_relationship (as relationships) <SET[1:?]>	PATH			group <- GROUPS<identification_assignment_relationship, 'relationships'> {CLASS_ID<'version relationship'>}
version_history to versionable_object (as versions) <SET[1:?]>	PATH			group <- (RELATE_GROUP_2_VO<representation, 'versions'>) (RELATE_GROUP_2_VO<product_definition, 'versions'>) (RELATE_GROUP_2_VO<property_definition, 'versions'>) (RELATE_GROUP_2_VO<shape_aspect, 'versions'>) (RELATE_GROUP_2_VO<product_category, 'versions', >) (RELATE_GROUP_2_VO<document, 'versions'>) (RELATE_GROUP_2_VO<product_definition_relationship, 'versions'>)

Application Element	AIM element	Source	Rules	Reference Path
				(RELATE_GROUP_2_VO<shape_aspect_relationship, 'versions'>)
VERSION_RELATIONSHIP	identification_assignment_relationship	41		((ROOT_CLASS<identification_assignment_relationship, 'version relationship'>))
version_relationship to versionable_object (predecessor)	PATH			identification_assignment_relationship identification_assignment_relationship.related_assignment -> (RELATE_ID_2_VO< representation >) (RELATE_ID_2_VO< product_definition >) (RELATE_ID_2_VO< property_definition >) (RELATE_ID_2_VO< shape_aspect >) (RELATE_ID_2_VO< product_category >) (RELATE_ID_2_VO< document >) (RELATE_ID_2_VO< product_definition_relationship >) (RELATE_ID_2_VO< shape_aspect_relationship >)
version_relationship to versionable_object (as successor)	PATH			identification_assignment_relationship identification_assignment_relationship.relate_assignment -> (RELATE_ID_2_VO< representation >) (RELATE_ID_2_VO< product_definition >) (RELATE_ID_2_VO< property_definition >) (RELATE_ID_2_VO< shape_aspect >) (RELATE_ID_2_VO< product_category >) (RELATE_ID_2_VO< document >) (RELATE_ID_2_VO< product_definition_relationship >) (RELATE_ID_2_VO< shape_aspect_relationship >)
reason	identification_assignment_relationship.description	41		
EVENT 1: VERSIONABLE_OBJECT_CHANGE_EVENT 2: APPROVAL_EVENT 3: CHECK	<i>see subtypes</i>			

Application Element	AIM element	Source	Rules	Reference Path
description	<i>see subtypes</i>			
event to date_and_time (as caused_when)	PATH			<i>see subtypes</i>
event to person_and_organisation (as caused_by)	PATH			<i>see subtypes</i>
CHANGE	action	41		{ROOT_CLASS<action, 'change'>}
class	action.description	41		
change to change_state (as possible_states) <SET[1:3] INV> 1: CHANGE_STATE = CHANGE_REQUEST 2: CHANGE_STATE = CHANGE_PLAN 3: CHANGE_STATE = CHANGE_REALISATION	PATH			1: (action action.chosen_method -> action_method <- action_request_solution.method action_request_solution action_request_solution.request -> versioned_action_request) 2: (action action.chosen_method -> action_method <- action_request_solution.method action_request_solution) 3: (action <- action_relationship.relatng_action action_relationship action_relationship.related_action -> action executed_action)
CHANGE_IMPACT	applied_action_request_assignment	218		applied_action_request_assignment <= action_request_assignment {ROOT_CLASS<applied_action_request_assignment, 'change impact'>}
change_impact to VERSIONABLE_OBJECT_change_event (as impact) <SET[1:?]>	PATH			applied_action_request_assignment applied_action_request_assignment.items [i] -> action_request_item

Application Element	AIM element	Source	Rules	Reference Path
				action_request_item = action {CLASS_ID<action, 'versionable object change event'>}
CHANGE_PLAN	action_request_solution	41		{[CLASS<action_request_solution, 'change plan', 'change state'>] [ROOT_CLASS<action_request_solution, 'change state'>]]}
change_plan to change_impact (as impact)	PATH			action_request_solution action_request_solution.request -> versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => applied_action_request_assignment {CLASS_ID<applied_action_request_assignment, 'change impact'>}
change_plan to change_REQUEST (as chosen_solution_for)	PATH			action_request_solution action_request_solution.request -> versioned_action_request {CLASS_ID<versioned_action_request, 'change request'>}
change_plan to check (as checks) <SET[0:?]>	PATH			action_request_solution action_request_solution = action_item action_item <- applied_action_assignment.items[i] applied_action_assignment action_assignment action_assignment.assigned_action -> action {CLASS_ID<action, 'check'>}
- from CHANGE_STATE				
description	action_request_solution.des cription	41		
change_PLAN to CHANGE (as change_reference)	PATH			action_request_solution action_request_solution.method ->

Application Element	AIM element	Source	Rules	Reference Path
				action_method <- action.chosen_method action {CLASS_ID<action, 'change'>}
change_PLAN to date_and_time (as date_time)	PATH			DAT_TIME_ASSGN<action_request_solution,'date time'>
change_PLAN to person_and_organisation (as author)	PATH			PERS_ORG_ASSGN<action_request_solution,'author'>
CHANGE_REALIZATION	executed_action	41		{[CLASS<executed_action, 'change realisation', 'change state'>] [ROOT_CLASS<executed_action, 'change state'>]}
change_realization to change_impact (as impact)	PATH			executed_action action_request_item = executed_action action_request_item <- applied_action_request_assignment.items[i] applied_action_request_assignment {CLASS_ID<applied_action_request_assignment, 'change impact'>}
change_REALISATION to change_PLAN (as realisation_of)	PATH			executed_action <= action action.chosen_method -> action_method <- action_request_solution.method action_request_solution {CLASS_ID<action_request_solution, 'change plan'>}
change_realization to check (as checks) <SET[0:?]>	PATH			executed_action executed_action = action_item action_item <- applied_action_assignment.items[i] applied_action_assignment action_assignment action_assignment.assigned_action -> action {CLASS_ID<action, 'check'>}

Application Element	AIM element	Source	Rules	Reference Path
- from CHANGE_STATE				
description	action.description	41		executed_action <= action action.description
change_REALISATION to CHANGE (as change_reference)	PATH			executed_action <= action <= action_relationship.related_action action_relationship action_relationship.relying_action action {CLASS_ID<action, 'change'>}
change_REALISATION to date_and_time (as date_time)	PATH			DAT_TIME_ASSGN<executed_action, 'date time'>
change_REALISATION to person_and_organisation (as author)	PATH			PERS_ORG_ASSGN<executed_action, 'author'>
CHANGE_REQUEST	versioned_action_request	41		{[CLASS<versioned_action_request, 'change request', 'change state'>] [ROOT_CLASS<versioned_action_request, 'change state'>]}
change_request to person_and_organisation (as initiator)	PATH			PERS_ORG_ASSGN<versioned_action_request, 'initiator'>
change_request to person_and_organisation (as addressee) <OPT>	PATH			PERS_ORG_ASSGN<versioned_action_request, 'addressee'>
problem	versioned_action_request.purpose	41		
solution_description <OPT>	action_request_solution.description	41		versioned_action_request <= action_request_solution.request action_request_solution action_request_solution.description
change_request to change_impact (as solution_alternatives) <SET[0:?]>	PATH			versioned_action_request <= action_request_assignment.assigned_action_request action_request_assignment => applied_action_request_assignment

Application Element	AIM element	Source	Rules	Reference Path
				{CLASS_ID<applied_action_request_assignment, 'change impact'>}
- from CHANGE_STATE				
description	versioned_action_request.description	41		
change_request to CHANGE (as change_reference)	PATH			versioned_action_request <- action_request_solution.request action_request_solution action_request_solution.method action_method <- action.chosen_method action {CLASS_ID<action, 'change'>}
change_REQUEST to date_and_time (as date_time)	PATH			DAT_TIME_ASSGN< versioned_action_request,'date time'>
change_REQUEST to person_and_organisation (as author)	PATH			PERS_ORG_ASSGN< versioned_action_request, 'author'>
CHANGE_STATE <ABS> 1: CHANGE_REQUEST 2: CHANGE_PLAN 3: CHANGE_REALIZATION	<i>see subtypes</i>			
change_state to change (as change_reference)	PATH			<i>see subtypes</i>
description	<i>see subtypes</i>			
change_state to date_and_time (as date_time)	PATH			<i>see subtypes</i>
change_state to person_and_organisation (as author)	PATH			<i>see subtypes</i>
CHECK	action	41		{[CLASS<action, 'check', 'event'>] [ROOT_CLASS<action, 'event'>]}
- from EVENT				
description	action.description	41		
check to date_and_time (as caused_when)	PATH			DAT_TIME_ASSGN<action,'caused when'>

Application Element	AIM element	Source	Rules	Reference Path
check to person_and_organisation (as caused_by)	PATH			PERS_ORG_ASSGN<action, 'caused by'>
VERSIONABLE_OBJECT_CHANGE_EVENT <ABS> 1: ENVISAGED_VERSION_CREATION 2: VERSION_CREATION 3: VERSION_MODIFICATION 4: VERSION_DELETION	1, 2, 3, 4: action	41		
- from EVENT				
description	action.description	41		
versionable_object_change_event to date_and_time (as caused_when)	PATH			DAT_TIME_ASSGN<action, 'caused when'>
versionable_object_change_event to person_and_organisation (as caused_by)	PATH			PERS_ORG_ASSGN<action, 'caused by'>
VERSION_CREATION	action	41		{[CLASS<action, 'version creation', 'versionable_object change event'>] [CLASS<action, 'versionable_object change event', 'event'>] [ROOT_CLASS<action, 'event'>]]}
version_creation to versionable_object (as base) <SET[0:?]>	PATH			action <- action_assignment.assigned_action action_assignment {action_assignment.role = 'base'} => (RELATE_ACT_2_VO< representation >) (RELATE_ACT_2_VO< product_definition >) (RELATE_ACT_2_VO< property_definition >) (RELATE_ACT_2_VO< shape_aspect >) (RELATE_ACT_2_VO< product_category >) (RELATE_ACT_2_VO< document >) (RELATE_ACT_2_VO< product_definition_relationship >) (RELATE_ACT_2_VO< shape_aspect_relationship >)
version_creation to versionable_object (as	PATH			action <-

Application Element	AIM element	Source	Rules	Reference Path
subject)				action_assignment.assigned_action action_assignment {action_assignment.role = 'subject'} => (RELATE_ACT_2_VO< representation >) (RELATE_ACT_2_VO< product_definition >) (RELATE_ACT_2_VO< property_definition >) (RELATE_ACT_2_VO< shape_aspect >) (RELATE_ACT_2_VO< product_category >) (RELATE_ACT_2_VO< document >) (RELATE_ACT_2_VO< product_definition_relationship >) (RELATE_ACT_2_VO< shape_aspect_relationship >)
- from EVENT				
description	action.description	41		
version_creation to date_and_time (as caused_when)	PATH			<i>see supertype</i>
version_creation to person_and_organisation (as caused_by)	PATH			<i>see supertype</i>
VERSION_DELETION	action	41		{[CLASS<action, 'version deletion', 'versionable_object change event'>] [CLASS<action, 'versionable_object change event', 'event'>] [ROOT_CLASS<action, 'event'>]]}
version_deletion to versionable_object (as subject)	PATH			action <- action_assignment.assigned_action action_assignment {action_assignment.role = 'subject'} => (RELATE_ACT_2_VO< representation >) (RELATE_ACT_2_VO< product_definition >) (RELATE_ACT_2_VO< property_definition >) (RELATE_ACT_2_VO< shape_aspect >) (RELATE_ACT_2_VO< product_category >) (RELATE_ACT_2_VO< document >)

Application Element	AIM element	Source	Rules	Reference Path
				(RELATE_ACT_2_VO< product_definition_relationship >) (RELATE_ACT_2_VO< shape_aspect_relationship >)
- from EVENT				
description	action.description	41		
version_deletion to date_and_time (as caused_when)	PATH			<i>see supertype</i>
version_deletion to person_and_organisation (as caused_by)	PATH			<i>see supertype</i>
VERSION_MODIFICATION	action	41		{[CLASS<action, 'version modification', 'versionable_object change event'>] [CLASS<action, 'versionable_object change event', 'event'>] [ROOT_CLASS<action, 'event'>]]}
version_modification to versionable_object (as base) <SET[0:?]>	PATH			action <- action_assignment.assigned_action action_assignment {action_assignment.role = 'base'} => (RELATE_ACT_2_VO< representation >) (RELATE_ACT_2_VO< product_definition >) (RELATE_ACT_2_VO< property_definition >) (RELATE_ACT_2_VO< shape_aspect >) (RELATE_ACT_2_VO< product_category >) (RELATE_ACT_2_VO< document >) (RELATE_ACT_2_VO< product_definition_relationship >) (RELATE_ACT_2_VO< shape_aspect_relationship >)
version_modification to versionable_object (as subject)	PATH			action <- action_assignment.assigned_action action_assignment {action_assignment.role = 'subject'} => (RELATE_ACT_2_VO< representation >) (RELATE_ACT_2_VO< product_definition >)

Application Element	AIM element	Source	Rules	Reference Path
				(RELATE_ACT_2_VO< property_definition >) (RELATE_ACT_2_VO< shape_aspect >) (RELATE_ACT_2_VO< product_category >) (RELATE_ACT_2_VO< document >) (RELATE_ACT_2_VO< product_definition_relationship >) (RELATE_ACT_2_VO< shape_aspect_relationship >)
- from EVENT				
description	action.description	41		
version_modification to date_and_time (as caused_when)	PATH			<i>see supertype</i>
version_modification to person_and_organisation (as caused_by)	PATH			<i>see supertype</i>
ENVISAGED_VERSION_CREATION	action	41		{[CLASS<action, 'envisaged version creation', 'versionable object change event'>] [CLASS<action, 'versionable_object change event', 'event'>] [ROOT_CLASS<action, 'event'>]]}
category	action.name	41		
envisaged_version_creation to versionable_object (as base) <SET[0:?]>	PATH			action <- action_assignment.assigned_action action_assignment {action_assignment.role = 'base'} => (RELATE_ACT_2_VO< representation >) (RELATE_ACT_2_VO< product_definition >) (RELATE_ACT_2_VO< property_definition >) (RELATE_ACT_2_VO< shape_aspect >) (RELATE_ACT_2_VO< product_category >) (RELATE_ACT_2_VO< document >) (RELATE_ACT_2_VO< product_definition_relationship >) (RELATE_ACT_2_VO< shape_aspect_relationship >)
- from EVENT				

Application Element	AIM element	Source	Rules	Reference Path
description	action.description	41		
envisaged_version_creation to date_and_time (as caused_when)	PATH			<i>see supertype</i>
envisaged_version_creation to person_and_organisation (as caused_by)	PATH			<i>see supertype</i>

Table 3 - Mapping table for definitions UoF

Application Element	AIM element	Source	Rules	Reference Path
DEFINITION (ABS) #1: design definition #2: general characteristics definition #3: functional definition #4: manufacturing definition #5: library definition #6: lightship definition #7: spacine table local coordinate system #8: ship material #9: plate renewal definition profile renewal definition	<i>See subtypes</i>			
definition to definable_object	See subtypes			

Application Element	AIM element	Source	Rules	Reference Path
(as defined_for)				
definition to global_ID (as id)	See subtypes			
definition to unit (as local_units)	<i>See subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	<i>See subtypes</i>
DEFINITION_RELATIONSHIP				
#1: if the definitions map to product_definition	#1: (product_definition_relation ship)	#1: 41		#1: {ROOT_CLASS<product_definition_relationship, 'definition relationship'>}
#2: if the definitions map to property_definition	#2: (property_definition_relatio nship)	#2: 45		#2: {ROOT_CLASS<property_definition_relationship, 'definition relationship'>}
#3: if the definitions map to representation	#3: (representation_relationship)	#3: 43		#3: {ROOT_CLASS<representation_relationship, 'definition relationship'>}
description	#1: (product_definition_- relationship.description) #2: (property_definition_- relationship.description) #3: (representation_relationship. description)	#1: 41 #2: 45 #3: 43		

Application Element	AIM element	Source	Rules	Reference Path
definition_relationship to definition (as definition_1)	<p>#1: (product_definition_relationship.relat- ing_product_definition)</p> <p>#2: (property_definition_relationship.rela- ting_property_definition)</p> <p>#3: (representation_relationship.rep_1)</p>			<p>#1: (product_definition_relationship product_definition_relationship.relat- ing_product_definition -> product_definition {product_definition.name = 'definition 1'} {CLASS_ID<product_definition, 'definition'>})</p> <p>#2: (property_definition_relationship property_definition_relationship.relat- ing_property_definition -> property_definition {property_definition.name = 'definition 1'} {CLASS_ID<property_definition, 'definition'>})</p> <p>#3: (representation_relationship representation.rep_1 -> representation {representation.name = 'definition 1'} {CLASS_ID<representation, 'definition'>})</p>
definition_relationship to definition (as definition_2)	#1: (product_definition_relation	#1: 41		#1:

Application Element	AIM element	Source	Rules	Reference Path
	ship.related_ product_definition) #2: (property_definition_relatio nship.related_property_defi nition) #3: (representation_relationship. rep_2)	#2: 41 #3: 43		(product_definition_relationship product_definition_relationship.related_product_ definition -> product_definition {product_definition.name = 'definition 2' } {CLASS_ID<product_definition, 'definition'>}) #2: (property_definition_relationship property_definition_relationship.related_property_ definition -> property_definition {property_definition.name = 'definition 2' } {CLASS_ID<property_definition, 'definition'>}) #3: (representation_relationship representation.rep_2 -> representation {representation.name = 'definition 2' } {CLASS_ID<representation, 'definition'>})
DESIGN_DEFINITION (ABS)	<i>See subtypes</i>			
#1: structural part design definition, system design definition,				

Application Element	AIM element	Source	Rules	Reference Path
hull cross section design definition structural part joint design definition #2: feature_design_definition weld design definition #3: library design definition #4: design load				
design_definition to representation (as representations) (S[0:?])	<i>See subtypes</i>			
- from DEFINITION				
design_definition to definable_object (as defined_for)	<i>See subtypes</i>			
design_definition to global_ID (as id)	<i>See subtypes</i>			
design_definition to unit (as local_units)	<i>See subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	#1: (VERSION_ID<extended_product_definition_shape>) #2: (VERSION_ID<representation>) #3: (VERSION_ID<extended_product_definition_shape>) (VERSION_ID<representation>) #4: (VERSION_ID<product_definition>)

Application Element	AIM element	Source	Rules	Reference Path
FUNCTIONAL_DEFINITION (ABS) #1: shiptype #2: edge cutout functional definition interior cutout functional definition edge feature functional definition #3: structural part functional definition structural system functional definition	<i>See subtypes</i>			
user_def_function	<i>See subtypes</i>			
- from DEFINITION				
functional_definition to definable_object (as defined_for)	<i>See subtypes</i>			
functional_definition to global_ID (as id)	<i>See subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	#1: (VERSION_ID<extended_product_related_product_category>) #2: (VERSION_ID<representation>)

Application Element	AIM element	Source	Rules	Reference Path
				#3: (VERSION<extended_property_definition>)
GENERAL_CHARACTERISTICS_- DEFINITION (ABS) #1: ship designation owner designation shipyard designation class and statutory designation class parameters principle characteristics freeboard characteristics global axis placement #2: regulation	<i>See subtypes</i>			
general_characteristics_definition to ship (as defined_for)	<i>See subtypes</i>			
- from DEFINITION				
general_characteristics_definition to global_id (as id)	See subtypes			
general_characteristics_definition to unit (as local_units)	<i>See subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	#1: (VERSION_ID<product_definition>) #2: (VERSION_ID<property_definition>)
LIBRARY_DEFINITION #1: If the subtypes map to a representation	#1: (representation) #2:	#1: 43 #2: 218	#1: 27 #2: 24	#1: ([CLASS<representation, 'library definition', 'definition'>])

Application Element	AIM element	Source	Rules	Reference Path
#2: If the subtypes map to a extended_product_definition_shape	(extended_product_definition_shape)			[CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]] #2: ([CLASS<extended_product_definition_shape, 'library definition', 'definition'> [CLASS<extended_product_definition_shape, 'definition', 'versionable object'> [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]))
library_definition to library_element_-reference (as source)	PATH		#1: 31 #2: 30	#1: (representation classification_item = representation classification_item <- applied_classification_assignment.items applied_classification_assignment) #2: (extended_product_definition_shape <= product_definition_shape classification_item = product_definition_shape classification_item <- applied_classification_assignment.items applied_classification_assignment)
- from DEFINITION				
library_definition to definable_object (as defined_for)	See subtypes			
library_definition to global_ID (as id)	#1: (representation.id) #3: (extended_product_definition_shape)		#1: 20, 22 #2: 20,	

Application Element	AIM element	Source	Rules	Reference Path
			26	
library_definition to unit (as local_units)	PATH			#1: (representation LOCAL_UNITS) #2: (extended_product_definition_shape <= product_definition_shape <= property_definition = presented_definition <= PDR -> representation LOCAL_UNITS)
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	#1: (VERSION_ID<representation>) #2: (VERSION_ID<extended_product_definition_shape>)
LIBRARY_DESIGN_DEFINITION #1: if the subtype maps to extended_product_definition_shape #2: if the subtype maps to representation	#1: (extended_product_definitio n_shape) #2: (representation)	#1: 218 #2: 41	#1: 32 #2: 33	#1: ({[CLASS<extended_product_definition_shape, 'library design definition', 'library definition'>] [CLASS<extended_product_definition_shape, 'library definition', 'definition'>] [CLASS<extended_product_definition_shape, 'library design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]})

Application Element	AIM element	Source	Rules	Reference Path
				#2: ({[CLASS<representation, 'linrary design definition', 'library definition'>] [CLASS<representation, 'library definition', 'definition'>] [CLASS<representation, 'linrary design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]})
- from LIBRARY_DEFINITION				
library_design_definition to library_element_reference (as source)	PATH		#1: 34 #2: 35	#1: (extended_product_definition_shape <= product_definition_shape classification_item = product_definition_shape classification_item <- applied_classification_assignment.items applied_classification_assignment) #2: (representation classification_item = representation classification_item <- applied_classification_assignment.items applied_classification_assignment)
- from DESIGN_DEFINITION				
library_design_definition to representation (as representations) (S[0:?])	#1: PATH #2: IDENTICAL MAPPING			#1: (extended_product_definition_shape <= product_definition_shape <= property_definition= represented_definition <- PDR ->

Application Element	AIM element	Source	Rules	Reference Path
				representation {representation.name = 'representations'}) #2: ({representation.name = 'representations'})
- from DEFINITION				
library_design_definition to global_id (as id)	#1: (extended_product_definition_shape.id) #2: (representation.id)	218 41	#1: 20, 26 #2: 20, 22	
library_design_definition to unit (as local_units) (S[1:?])	PATH			#1: (extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR -> representation LOCAL_UNITS) #2: (representation LOCAL_UNITS)
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	(VERSION_ID<extended_product_definition_shape>) (VERSION_ID<representation>)
LIBRARY_MANUFACTURING_- DEFINITION	extended_product_definition_shape	218	23	{[CLASS<extended_product_definition_shape, 'library manufacturing definition', 'library definition'>] [CLASS<extended_product_definition_shape, 'library definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]
- from LIBRARY_DEFINITION				
library_manufacturing_definition to library_element_reference (as source)	PATH		25	extended_product_definition_shape <= product_definition_shape CLASS_ROLE<product_definition_shape, 'source'>
- from MANUFACTURING_DEFINITION				
library_manufacturing_definition to representation (as representations)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR -> representation {representation.name = 'representations'}
- from DEFINITION				
library_manufacturing_definition to global_id (as id)	extended_product_definition_shape.id		20, 21	
library_manufacturing_definition to unit (as local_units) (S[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR -> representation LOCAL_UNITS
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	28	VERSION_ID<extended_product_definition_shape>
MANUFACTURING_DEFINITION (ABS)	extended_product_definition_shape	218	29	{[CLASS<extended_product_definition_shape, 'manufacturing definition', 'definition'] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape,

Application Element	AIM element	Source	Rules	Reference Path
				'versionable object'>]}
manufacturing_definition to representation (as representations)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition= represented_definition <= PDR -> representation {representation.name = 'representations'}
- from DEFINITION				
manufacturing_definition to global_id (as id)	extended_product_definition_shape.id	218	20, 26	
manufacturing_definition to unit (as local_units) (S[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<manufacturing definition parameters> -> representation LOCAL_UNITS
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	28	VERSION_ID<extended_product_definition_shape>

Table 4 - Mapping table for design_loads UoF

Application Element	AIM element	Source	Rules	Reference Path
BENDING_MOMENT (ABS)	product_definition	41		See subtype mapping
DESIGN_LOAD (ABS)	product_definition	41		See subtype mapping
design_load to ship (as defined_for) (S[1:?])	PATH			product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product
DESIGN_STILL_WATER_BENDING_MOMENT	product_definition	41	43	{[CLASS<product_definition, 'design still water bending moment', 'bending moment'>] [CLASS<product_definition, 'bending moment', 'design load'>] [ROOT_CLASS<product_definition, 'design load'>]]}
hogging_amidship	value_representation_item. value_component	43	36, 37	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition {extended_property_definition <= property_definition} property_definition <- PDR<'design still water bending moment parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'hogging amidship',mass_measure>
sagging_amidship	value_representation_item. value_component	43	36, 37	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition {extended_property_definition <= property_definition} property_definition <- PDR<'design still water bending moment parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'hogging amidship',mass_measure>
- from DESIGN_LOAD				

Application Element	AIM element	Source	Rules	Reference Path
design_still_water_bending_moment to ship (as defined for) (S[1:?])	<i>See supertype</i>			
- from DESIGN_DEFINITION				
design_still_water_bending_moment to shape_representation (as representations) (S[0:?])	<i>See supertype</i>			
- from DEFINITION				
design_still_water_bending_moment to global_id (as id)	<i>See supertype</i>			
design_still_water_bending_moment to unit (as local_units) (S[0:?])	<i>See supertype</i>			
- from VERSIONAL_OBJECT				
version_id	<i>See supertype</i>			
DESIGN_STILL_WATER_SHEAR_FORCE	product_definition	41	44	{[CLASS<product_definition, 'design still water shear force', 'shear force'>] [CLASS<product_definition, 'shear force', 'design load'>] [ROOT_CLASS<product_definition, 'design load'>]]}
design_still_water_shear_force to desing_SWSF_values (as table_of_shear_force_values) (SET[1:?])	PATH		40	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition <- PDR -> representation {representation.name = 'table of shear force value'} representation.items[i] -> representation_item {representation_item.name = 'design SWSF values'} representation_item => compound_representation_item
- from DESIGN_LOAD				
design_still_water_bending_moment to ship (as defined for) (S[1:?])	<i>See supertype</i>			
- from DESIGN_DEFINITION				
design_still_water_bending_moment to shape_representation (as representations)	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(S[0:?])				
- from DEFINITION				
design_still_water_bending_moment to global_id (as id)	<i>See supertype</i>			
design_still_water_bending_moment to unit (as local_units) (S[0:?])	<i>See supertype</i>			
- from VERSIONAL_OBJECT				
version_id	<i>See supertype</i>			
DESIGN_VERTICAL_WAVE_BENDING_MOMENT	product_definition	41	45	{[CLASS<product_definition, 'design vertical wave bending moment', 'bending moment'>] [CLASS<product_definition, 'bending moment', 'design load'>] [ROOT_CLASS<product_definition, 'desing load'>]]}
hogging	value_representation_item. value_component	43	36, 38	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition {extended_property_definition <= property_definition} property_definition <- PDR<'design still water bending moment parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'hogging amidship',mass_measure>
sagging	value_representation_item. value_component	43	36, 38	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition {extended_property_definition <= property_definition} property_definition <- PDR<'design still water bending moment parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'hogging amidship',mass_measure>
- from DESIGN_LOAD				
design_still_water_bending_moment to ship (as defined for) (S[1:?])	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from DESIGN_DEFINITION				
design_still_water_bending_moment to shape_representation (as representations) (S[0:?])	See supertype			
- from DEFINITION				
design_still_water_bending_moment to global_id (as id)	See supertype			
design_still_water_bending_moment to unit (as local_units) (S[0:?])	See supertype			
- from VERSIONAL_OBJECT				
version_id	See supertype			
DESIGN_VERTICAL_WAVE_SHEAR_FORCE	product_definition	41	46	{[CLASS<product_definition, 'design vertical wave shear force', 'shear force'>] [CLASS<product_definition,'shear force', 'design load'>] [ROOT_CLASS<product_definition,'design load'>]]}
negative_value	value_representation_item. value_component	43	36, 39	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition {extended_property_definition <= property_definition} property_definition <- PDR<'design still water bending moment parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'hogging amidship',mass_measure>
positive_value	value_representation_item. value_component	43	36, 39	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition {extended_property_definition <= property_definition} property_definition <- PDR<'design still water bending moment parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'hogging amidship',mass_measure>

Application Element	AIM element	Source	Rules	Reference Path
- from DESIGN_LOAD				
design_still_water_bending_moment to ship (as defined for) (S[1:?])	<i>See supertype</i>			
- from DESIGN_DEFINITION				
design_still_water_bending_moment to shape_representation (as representations) (S[0:?])	<i>See supertype</i>			
- from DEFINITION				
design_still_water_bending_moment to global_id (as id)	<i>See supertype</i>			
design_still_water_bending_moment to unit (as local_units) (S[0:?])	<i>See supertype</i>			
- from VERSIONAL_OBJECT				
version_id	<i>See supertype</i>			
DESIGN_SWSF_VALUES	compound_representation_item	43		
negative_value	value_representation_item. value_component	43	36, 49	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [1] -> VAL_REP_ITEM_CD<'negative value', 'force unit'>
positive_value	value_representation_item. value_component	43	36, 49	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [2] -> VAL_REP_ITEM_CD<'positive value', 'force unit'>
design_SWSF_values to spacing_position (as position)	PATH		49	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [3] -> representation_item =>

Application Element	AIM element	Source	Rules	Reference Path
				{representation_item.name = 'position'} compound_representation_item CLASS_ID<compound_representation_item, 'longitudinal position'>
MAXIMUM_PERMISSIBLE_SWBM_VALUES	compound_representation_item	43		
hogging	value_representation_item. value_component	43	36, 50	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [1] -> VAL_REP_ITEM_CD<'positive value', 'force unit'>
sagging	value_representation_item. value_component	43	36, 50	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [2] -> VAL_REP_ITEM_CD<'positive value', 'force unit'>
maximum_permissible_SWBM_values to spacing_position (as position)	PATH		50	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [3] -> representation_item => {representation_item.name = 'position'} compound_representation_item CLASS_ID<compound_representation_item, 'longitudinal position'>
MAXIMUM_PERMISSIBLE_SWSF_VALUES	compound_representation_item	43		
negative_value	value_representation_item. value_component	43	36, 51	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [1] -> VAL_REP_ITEM_CD<'positive value', 'force unit'>
positive_value	value_representation_item.	43	36, 51	compound_representation_item

Application Element	AIM element	Source	Rules	Reference Path
	value_component			compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [2] -> VAL_REP_ITEM_CD<'positive value', 'force unit'>
maximum_permissible_SWSF_values to spacing_position (as position)	PATH		51	compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item [3] -> representation_item => {representation_item.name = 'position'} compound_representation_item CLASS_ID<compound_representation_item, 'longitudinal position'>
MAXIMUM_PERMISSIBLE_STILL_ WATER_BENDING_MOMENT	product_definition	41	47	{[CLASS<product_definition, 'maximum permissible still water bending moment', 'bending moment'>] [CLASS<product_definition, 'bending moment', 'design load'>] [ROOT_CLASS<product_definition, 'design load'>]]}
maximum_permissible_still_water_bending_ moment to maximum_permissible_swbm_values (as table_of_max_bending_moment_values) (SET[1:?])	PATH		41	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition <- PDR<'maximum permissible still water bending moment va lues'> -> representation {representation.name = 'table of moment value'} representation.items[i] -> representation_item {representation_item.name = 'maximum permissible SWBM values'} representation_item => compound_representation_item
- from DESIGN_LOAD				
design_still_water_bending_moment to ship (as defined for) (S[1:?])	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from DESIGN_DEFINITION				
design_still_water_bending_moment to shape_representation (as representations) (S[0:?])	<i>See supertype</i>			
- from DEFINITION				
design_still_water_bending_moment to global_id (as id)	<i>See supertype</i>			
design_still_water_bending_moment to unit (as local_units) (S[0:?])	<i>See supertype</i>			
- from VERSIONAL_OBJECT				
version_id	<i>See supertype</i>			
MAXIMUM_PERMISSIBLE_STILL_WATER_SHEAR_FORCE	Product_definition	41	36, 48	{[CLASS<product_definition, 'maximum permissible still water shear force','shear force'>]] [CLASS<product_definition, 'shear force', 'design load'>] [ROOT_CLASS<product_definition, 'design load'>]]}
maximum_permissible_still_water_shear_force to maximum_permissible_swsf_values (as table_of_max_shear_force_values) (SET[1:?])	PATH		42	product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition <- PDR<'maximum permissible still water shear force values' > -> representation {representation.name = 'table of shear force value'} representation.items[i] -> representation_item {representation_item.name = 'maximum permissible SWSF values'} representation_item => compound_representation_item
- from DESIGN_LOAD				
design_still_water_bending_moment to ship (as defined for) (S[1:?])	<i>See supertype</i>			
- from DESIGN_DEFINITION				

Application Element	AIM element	Source	Rules	Reference Path
design_still_water_bending_moment to shape_representation (as representations) (S[0:?])	<i>See supertype</i>			
- from DEFINITION				
design_still_water_bending_moment to global_id (as id)	<i>See supertype</i>			
design_still_water_bending_moment to unit (as local_units) (S[0:?])	<i>See supertype</i>			
- from VERSIONAL_OBJECT				
version_id	<i>See supertype</i>			
SHEAR_FORCE	product_definition	41		<i>See subtype mapping</i>

Table 5 - Mapping table for UoF external_references

Application Element	AIM element	Source	Rules	Reference Path
EXTERNAL_REFERENCE	applied_external_identification_assignment	218		applied_external_identification_assignment <= external_identification_assignment <= ID_ROLE<'external reference'>
description	identification_role.description	41		applied_external_identification_assignment <= external_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.description
location	external_identification_assignment.source	41		applied_external_identification_assignment <= external_identification_assignment external_identification_assignment.source -> external_source {(CLASS_ID<external_source, 'universal resource locator'>)} (CLASS_ID<external_source, 'address'>))}
UNIVERSAL_RESOURCE_LOCATOR	external_source	41	52	{ROOT_CLASS<external_source, 'universal resource locator'>}
location	external_source.source_id	41	53	external_source external_source.source_id -> source_item {source_item = identifier}
machine_address	external_source.source_id	41	53	external_source external_source.source_id -> source_item {source_item = identifier}
other_protocol_type	external_source.source_id	41	53	external_source external_source.source_id -> source_item {source_item = identifier}

Application Element	AIM element	Source	Rules	Reference Path
port	external_source.source_id	41	53	external_source external_source.source_id -> source_item {source_item = identifier}
protocol	external_source.source_id	41	53	external_source external_source.source_id -> source_item {source_item = identifier}
ADDRESS	external_source	41	58, 59, 62	{ROOT_CLASS<external_source, 'address'>}
internal_location <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'internal location'>
street_number <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'street number'>
street <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'street' >
postal_box <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'postal box' >
town <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'town' >
region <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'region' >
postal_code <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'postal code' >
country <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'country' >
facsimile_number <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'facsimile number' >
telephone_number <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'telephone number' >
electronic_mail_address <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'electronic mail address' >
telex_number <OPTIONAL>	external_source.source_id	41		EXT_SRC_REL< 'telex number' >
EXTERNAL_INSTANCE_REFERENCE	applied_external_identification_assignment	218		applied_external_identification_assignment <= external_identification_assignment <= ID_ROLE<'external instance reference'> external_identification_assignment external_identification_assignment.source -> ROOT_CLASS<external_source, 'instance address'>
schema_name	external_source.source_id	41		applied_external_identification_assignment <= external_identification_assignment external_identification_assignment.source -> EXT_SRC_REL< 'schema name' >

Application Element	AIM element	Source	Rules	Reference Path
entity_type	external_source.source_id	41		applied_external_identification_assignment <= external_identification_assignment external_identification_assignment.source -> EXT_SRC_REL< 'entity type' >
external_instance_reference to global_id (as target_guid)	PATH	41	61	applied_external_identification_assignment <= external_identification_assignment <= identification_assignment identification_assignment.assigned_id
DOCUMENT	document	41		{[CLASS<document, 'document', 'versionable object'>] [ROOT_CLASS<document, 'versionable object'>]}
#1: document to person (as has_author) #2: document to organisation (as has_author) #3: document to person_and_organization (as has_author)	#1: person_assignment.assigned _person #2: organization_assignment.ass igned_organization #3: person_and_organization_as signment.assigned_ person_and_organization	41		1: PERS_ASSGN<document, 'document author'> 2: ORG_ASSGN<document, 'document author'> 3: PERS_ORG_ASSGN<document, 'document author'>
has_title	document.name	41		
source_type	document_type	41		document document.kind -> document_type
summary <OPTIONAL>	document.description	41		
document to document_usage_constraint (as document_subset) <SET[1:?]> <INVERSE>	PATH			document <- document_usage_constraint.source document_usage_constraint
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			VERSION_ID< document >
DOCUMENT_REFERENCE	applied_document_reference	218		applied_document_reference <= document_reference

Application Element	AIM element	Source	Rules	Reference Path
				{ document_reference document_reference.role -> object_role object_role.name = 'document reference'}
document_reference to document_referent (as assigned_document)	PATH			applied_document_reference <= document_reference document_reference.assigned_document -> (document) (document <- document_usage_constraint.source document_usage_constraint)
DOCUMENT_REFERENCE_WITH_ADDRES S	[applied_document_referenc e] [applied_external_identificat ion_assignment]	218	74	{ applied_document_reference <= document_reference { document_reference document_reference.role -> object_role object_role.name = 'document reference'} external_identification_item = document_reference external_identification_item <- applied_external_identification_assignment.items[i] applied_external_identification_assignment <= external_identification_assignment <= ID_ROLE<'external reference'>}
- from DOCUMENT_REFERENCE				
document_reference_with_address to document_referent (as assigned_document)	PATH			<i>see supertype</i>
- from EXTERNAL_REFERENCE				
description	<i>see supertype</i>			
location	<i>see supertype</i>			
DOCUMENT_USAGE_CONSTRAINT	document_usage_constraint	41	60	
element_name	document_usage_constraint. subject_element	41		

Application Element	AIM element	Source	Rules	Reference Path
line_number <OPTIONAL>	identification_assignment.as signed_id	41		document_usage_constraint HAS_ID< document_usage_constraint, 'line number'> identification_assignment.assigned_id
page <OPTIONAL>	identification_assignment.as signed_id	41		document_usage_constraint HAS_ID< document_usage_constraint, 'page'> identification_assignment.assigned_id
paragraph <OPTIONAL>	identification_assignment.as signed_id	41		document_usage_constraint HAS_ID< document_usage_constraint, 'paragraph'> identification_assignment.assigned_id
section <OPTIONAL>	identification_assignment.as signed_id	41		document_usage_constraint HAS_ID< document_usage_constraint, 'section'> identification_assignment.assigned_id
document_usage_constraint to document (as source)	PATH			document_usage_constraint document_usage_constraint.source -> document
BSU (ABS) #1: CLASS_BSU #2: PROPERTY_BSU #3: SUPPLIER_BSU	1: externally_defined_class 2: externally_defined_general_ property 3: organization	#1: 218 #2: 218 #3: 41		<i>see subtypes</i>
code	<i>see subtypes</i>			
BOOLEAN_PROPERTY_VALUE	representation	43		representation {representation.name='property value' }
val	value_representation_item.v alue_component	43		representation representation.items[i] -> VAL_REP_ITEM< '.UNUSED.', descriptive_measure > {(value_representation_item.value_component = 'true') (value_representation_item.value_component = 'false')}
- from PROPERTY_VALUE				
boolean_property_value to property_BSU (as property_identifier)	PATH			<i>see supertype</i>
CLASS_BSU	externally_defined_class	218		[externally_defined_class <=

Application Element	AIM element	Source	Rules	Reference Path
				externally_defined_item] [externally_defined_class <= class]
version	identification_assignment.as signed_id	41	54, 56, 65, 66, 67	externally_defined_class externally_defined_class = identification_item identification_item <= library_class_version_assignment.items[i] library_class_version_assignment <= applied_external_identification_assignment <= external_identification_assignment <= identification_assignment identification_assignment.assigned_id { ID_ROLE<'version'> }
class_BSU to supplier_BSU (as delivered_by)	PATH		64	externally_defined_class <= externally_defined_item externally_defined_item.source -> external_source => ORG_ASSGN<known_source, 'library supplier'>
absolute_id <DER>				
dic_identifier <DER>				
- from BSU				
code	externally_defined_item.ite m_id	41	55	externally_defined_class <= externally_defined_item externally_defined_item.item_id
INTEGER_PROPERTY_VALUE	representation	43		Representation {representation.name='property value' }
val	value_representation_item.v alue_component	43		Representation representation.items[i] -> VAL_REP_ITEM< '.UNUSED.', numeric_measure >
- from PROPERTY_VALUE				
integer_property_value to property_BSU (as property_identifier)	PATH			<i>see supertype</i>

Application Element	AIM element	Source	Rules	Reference Path
LIBRARY_ELEMENT_REFERENCE	applied_classification_assignment		73	
library_element_reference to class_BSU (as library_identifier)	PATH			<pre> applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'definitional class membership'} classification_assignment.assigned_classification -> group => {group.name = 'library identifier'} class => externally_defined_class </pre>
library_element_reference to property_value (as property_value_pairs) <SET[1:?]>	PATH		72	<pre> applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'definitional class membership'} classification_assignment.assigned_classification -> group => {group.name = 'library identifier'} class => externally_defined_class <= externally_defined_item <- externally_defined_item_relationship.related_item {externally_defined_item_relationship.name='name scope '} externally_defined_item_relationship externally_defined_item_relationship.relateing_item externally_defined_item => externally_defined_general_property <= general_property <- general_property_association.base_definition general_property_association </pre>

Application Element	AIM element	Source	Rules	Reference Path
				{ general_property_association.name = 'definitional' } general_property_association.derived_definition -> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation { representation.name='property value' }
LOGICAL_PROPERTY_VALUE	representation	43		
val	value_representation_item	43		Representation representation.items[i] -> VAL_REP_ITEM< 'UNUSED. ', descriptive_measure > { (value_representation_item.value_component = 'true') (value_representation_item.value_component = 'false') (value_representation_item.value_component = 'unknown') }
- from PROPERTY_VALUE				
logical_property_value to property_BSU (as property_identifier)	<i>see supertype</i>			
NUMBER_PROPERTY_VALUE	representation	43		Representation { representation.name='property value' }
val	value_representation_item.v alue_component	43		Representation representation.items[i] -> VAL_REP_ITEM< 'UNUSED.', numeric_measure >
- from PROPERTY_VALUE				
nummer_property_value to property_BSU (as property_identifier)	<i>see supertype</i>			
PROPERTY_BSU	externally_defined_general_ property	218		[externally_defined_general_property <= externally_defined_item] [externally_defined_general_property <= general_property]
version	identification_assignment.as	41	54, 71,	externally_defined_general_property

Application Element	AIM element	Source	Rules	Reference Path
	signed_id		69, 70	externally_defined_general_property = identification_item identification_item <- library_property_version_assignment.items[i] library_property_version_assignment <= external_identification_assignment <= identification_assignment identification_assignment.assigned_id {ID_ROLE<'version'>}
PROPERTY_BSU to CLASS_BSU (as name_scope)	PATH		68	externally_defined_general_property <= externally_defined_item <- externally_defined_item_relationship.relating_item externally_defined_item_relationship {externally_defined_item_relationship.name='name scope' '} externally_defined_item_relationship.related_item -> externally_defined_item => externally_defined_class
dic_identifier <DER>				
- from BSU				
code	externally_defined_item.item_id	41		externally_defined_general_property <= externally_defined_item externally_defined_item.item_id
PROPERTY_VALUE (ABS)	representation	43		Representation {representation.name='property value' }
property_value to Property_BSU (as property_identifier)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition <- general_property_association.derived_definition general_property_association {general_property_association.name='definitional' } general_property_association.base_definition ->

Application Element	AIM element	Source	Rules	Reference Path
				general_property => externally_defined_general_property
REAL_PROPERTY_VALUE	representation	43		representation {representation.name='property value' }
val	value_representation_item.v alue_component	43		representation representation.items[i] -> VAL_REP_ITEM< '.UNUSED.', numeric_measure >
- from PROPERTY_VALUE				
real_property_value to property_BSU (as property_identifier)	PATH			<i>see supertype</i>
STRING_PROPERTY_VALUE	representation	43		representation {representation.name='property value' }
val	value_representation_item.v alue_component	43		representation representation.items[i] -> VAL_REP_ITEM< '.UNUSED.', descriptive_measure >
- from PROPERTY_VALUE				
string_property_value to property_BSU (as property_identifier)	PATH			<i>see supertype</i>
SUPPLIER_BSU	organization	41		{organization.name = '.UNUSED.' }
- from BSU				
code	organization.id		57	organization organization.id

Table 6 - Mapping table for hull_cross_sections UoF

Application Element	AIM element	Source	Rules	Reference Path
FLARE_AREA	shape_aspect	41	87, 88, 89, 90	{ROOT_CLASS<shape_aspect, 'flare area buckling'>}
area	value_representation_item.value_component	43		shape_aspect <- PDR<'flare area parameters'> -> representation representation.items [i] -> VAL_REP_ITEM<'area', area_measure>
height	value_representation_item.value_component	43		shape_aspect <- PDR<'flare area parameters'> -> representation representation.items [i] -> VAL_REP_ITEM<'height', positive_length_measure>
HULL_CROSS_SECTION	[product_definition] [group]	41	75, 76, 77, 78	{[CLASS<product_definition, 'hull cross section', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>] [CLASS<product_definition, 'hull cross section', 'item structure'>] [CLASS<product_definition, 'item structure', 'definable object'>] [CLASS<product_definition, 'item structure', 'versionable object'>] [ROOT_CLASS<product_definition, 'versionable object'>] [product_definition LINK_TO_GROUP<product_definition>]}
- from ITEM_STRUCTURE				
hull_cross_section to external_instance_reference (as external_items) (S[0:?])	PATH			group <- GROUPS<product_definition, 'item structure'> {(EXT_INST_REF<product_definition, 'ship structures', 'compartment'>) (EXT_INST_REF<product_definition, 'ship structures', 'structural system'>)}
hull_cross_section to external_instance_reference (as external_relationships) (S[0:?])	- (relationships are not valid; set shall be zero)		75	

Application Element	AIM element PATH	Source	Rules	Reference Path
hull_cross_section to item (as items) (S[0:?])				group <- GROUPS<product_definition, 'item structure' {(CLASS_ID<product_definition, 'compartment') (CLASS_ID<product_definition, 'structural system')}>
hull_cross_section to item_relationship (as relationships) (S[0:?])	- (relationships are not valid; set shall be zero)		75	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	83	product_definition <- HAS_ID<product_definition, 'version id' identification_assignment.assigned_id
- from ITEM				
name	product_definition.name	41		
hull_cross_section to external_reference (as documentation) (SET[0:?])	PATH			product_definition document_reference_item = product_definition document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = 'UNUSED. ']}
description (OPT)	product_definition.description	41		
hull_cross_section to ship (as ship_context) (OPT)	PATH			product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structure' product_definition_formation.of_product -> product {CLASS_ID<product, 'ship'>}}
- from DEFINABLE_OBJECT				
hull_cross_section to global_id (as id)	product_definition.id	41	81	

Application Element	AIM element PATH	Source 2.	Rules	Reference Path
hull_cross_section to definition (as definitions) (INVERSE) (SET[0:?])				product_definition characterized_definition = product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape => extended_product_definition_shape {CLASS_ID<extended_product_definition_shape, 'hull cross section design definition'>}
HULL_CROSS_SECTION_DESIGN_DEFINITION	extended_product_definition_s hape	218	79, 82, 89	{extended_product_definition_shape <= product_definition_shape} {[CLASS<extended_product_definition_shape, 'hull cross section design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
mirrored_symmetry	descriptive_representation_item .description	45	85, 86	extended_product_definition_shape <= product_definition_shape <= property_definition represented_definition = property_definition represented_definition <- PDR<'hull cross section design definition parameters'> -> representation representation.items [i] -> descriptive_representation_item {descriptive_representation_item.name = 'mirrored symmetry'} {(descriptive_representation_item.description = 'yes') (descriptive_representation_item.description = 'no')}
hull_cross_section_design_definition to flare_area (as flare_area_buckling)	PATH			extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape

Application Element	AIM element	Source	Rules	Reference Path
				{CLASS_ID<shape_aspect, 'flare area buckling'>}
hull_cross_section_design_definition to hull_cross_section (as defined_for)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = product_definition {CLASS_ID<product_definition, 'hull cross section'>}
- from DESIGN_DEFINITION				
hull_cross_section_design_definition to representation (as representations) (SET[0:?])	PATH			extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect SA2REP_D<'hull cross section shape representation'> representation {representation.name = 'representations'} => shape_representation { => (edge_based_wireframe_shape_representation) (non_manifold_surface_shape_representation)}
- from DEFINITION				
hull_cross_section_design_definition to global_id (as id)	extended_product_definition_s hape.id	218	84	
hull_cross_section_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	83	extended_product_definition_shape <= HAS_ID<extended_product_definition_shape, 'version id'> identification_assignment.assigned_id

Table 7 - Mapping table for items UoF

Application Element	AIM element	Source	Rules	Reference Path
DEFINABLE_OBJECT (ABS) #1: item #2: item_relationship #3: item_structure	<i>See subtypes</i>			
definable_object to definition (as definitions) (INV)	<i>See subtypes</i>			
definable_object to global_id (as id)	<i>See subtypes</i>			
GLOBAL_ID	identifier	41		
company_id	get_company_id(identifier)	218		
local_id	get_local_id(identifier)	218		
ITEM (ABS) #1: ship #2: part, system, assembly, hull cross section, structural part connection implementation #3: feature	<i>See subtypes</i>			
description	<i>See subtypes</i>			
name	<i>See subtypes</i>			
item to external_reference (as documentation)	<i>See subtypes</i>			
item to ship (as ship_context)	<i>See subtypes</i>			
- from DEFINABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
item to definition (as definitions) (INV)	See subtypes			
item to global_id (as id)	See subtypes			
ITEM_RELATIONSHIP (ABS) #1: part relationship assembly relationship system relationship structural part joint #2: feature relationship structural part penetration relationship structural system penetration relationship	<i>See subtypes</i>			
contxt	<i>See subtypes</i>			
item_relationship to external_instance_reference (as external_item_1) (OPT)	<i>See subtypes</i>			
item_relationship to external_instance_reference (as external_item_2) (OPT)	<i>See subtypes</i>			
item_relationship to item (as item_1) (OPT)	<i>See subtypes</i>			
item_relationship to item (as item_2) (OPT)	<i>See subtypes</i>			
- from DEFINABLE_OBJECT				
item_relationship to definition (as definitions) (INV)	<i>See subtypes</i>			
item_relationship to global_id (as id)	<i>See subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.	41		<i>See subtypes</i>

Application Element	AIM element	Source	Rules	Reference Path
	assigned_id			
ITEM_STRUCTURE (ABS)	group	41		<i>See subtypes</i>
item_structure to external_instance_reference (as external_items) (S[0:?])	<i>See subtypes</i>			
item_structure to external_instance_reference (as external_relationships) (S[0:?])	<i>See subtypes</i>			
item_structure to item (as items) (S[0:?])	<i>See subtypes</i>			
item_structure to item_relationship (as relationships) (S[0:?])	<i>See subtypes</i>			
- from DEFINABLE_OBJECT				
item_structure to definition (as definitions) (INV)	<i>See subtypes</i>			
item_structure to global_id (as id)	<i>See subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	94	VERSION_ID<group>
SHIP	product	41		{[CLASS<product, 'ship', 'item'>] [CLASS<product, 'item', 'definable object'>] [ROOT_CLASS<product, 'definable object'>]}
ship to item (as ship_items) (INV) (S[0:?]) #1: if the item is a part, a system, an assembly, a hull_cross_section, a structural_part_connection_implementation #2: if the item is a feature	#1: PATH #2: PATH			#1: (product <- product_definition_formation.of_product product_definition_formation <- product_definition.formation product_definition {product_definition.description = 'ship items'}) #2: (product <- product_definition_formation.of_product

Application Element	AIM element	Source	Rules	Reference Path
				<pre> product_definition_formation <- product_definition.formation product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => extended_shape_aspect {extended_shape_aspect.description = 'ship items'}) </pre>
ship to unit (as units)	PATH (S[0:?])			<pre> product <- product_definition_formation.of_product product_definition_formation <- product_definition.formation product_definition = characterised_product_definition = characterised_product_definition = characterised_definition <- property_definition.definition property_definition <- PDR -> representation representation.context_of_items -> representation_context {[representation_context.context_type = 'units'] [representation_context.context_idenfifier = '.UNUSED.']} => global_unit_assigned_context global_unit_assigned_context.units -> unit </pre>
- from ITEM				

Application Element	AIM element	Source	Rules	Reference Path
name	product.name	41		
description	product.description	41		
ship to external_reference (as documentation)) (S[1:?])	PATH			<pre> product = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = '.UNUSED. ']} </pre>
- from DEFINABLE_OBJECT				
ship to definition (as definitions) (INV)	PATH			<pre> product <- product_definition_formation.of_product product_definition_formation <- product_definition.formation product_definition </pre>
ship to global_id (as id)	product.id	41	91, 92	
VERSIONABLE_OBJECT (ABS)	<i>See subtypes</i>			
#1: definition #2: item_relationship #3: item_structure				
version_id	identification_assignment. assigned_id	41		<i>See subtypes</i>

Table 8 - Mapping table for location_concepts UoF

Application Element	AIM element	Source	Rules	Reference Path
GLOBAL_AXIS_PLACEMENT	[product_definition] [axis2_placement_3d]	41 42	95, 96, 97	{[CLASS<product_definition, 'global axis placement', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>] PROD_DEF_TO_REP<'global axis placement'> representation {representation.name = 'global axis representation' representation.context_of_items representation_context => {representation_context.context_type = 'global co-ordinate space' geometric_representation_context {geometric_representation_context.coordinate_space_dimension = 3} REP_ITEM<'global axes and origin'> geometric_representation_item => placement => axis2_placement_3d
after_perpendicular_offset	value_representation_item.value_component	43	96, 97	PROD_DEF_TO_REP<'global axis placement'> representation {representation.name = 'global axis representation' representation.context_of_items representation_context => {representation_context.context_type = 'global co-ordinate space' geometric_representation_context REP_TO_VAL_REP_ITEM<'after perpendicular offset', length_measure>
orientation	descriptive_representation_item.description	45	96, 97	PROD_DEF_TO_REP<'global axis placement'> representation {representation.name = 'global axis representation' representation.context_of_items representation_context =>

Application Element	AIM element	Source	Rules	Reference Path
				{representation_context.context_type = 'global co-ordinate space'} geometric_representation_context REP_ITEM<'orientation'> descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description = 'forward pointing') (descriptive_representation_item.description = 'aft pointing')}
-from GENERAL_CHARACTERISTICS_DEFINITION				
global_axis_placement to ship (as defined_for) <SET[1:?]>	PATH			PROD_DEF_PRODUCT {CLASS_ID<product, 'ship'>}
-from DEFINITION				
global_axis_placement to global_id (as id)	product_definition.id	41	99, 100	
global_axis_placement to unit (as local_units) <SET>	PATH			PROD_DEF_TO_UNITS<'global axis placement'>
-from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	98	VERSION_ID<product_definition>
LOCAL_CO_ORDINATE_SYSTEM	[extended_property_definition] [axis2_placement_3d]	216 42	101, 102, 97	{[CLASS<extended_property_definition, 'local co-ordinate system', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]} PROP_DEF_REP_HELP<'local co-ordinate system'> representation {representation.name = 'local axis representation'} representation.context_of_items representation_context => {representation_context.context_type = 'local co-ordinate space'} geometric_representation_context

Application Element	AIM element	Source	Rules	Reference Path
				{geometric_representation_context.coordinate_space_dimension = 3} REP_ITEM<'local axes and origin' geometric_representation_item => placement => axis2_placement_3d
parent <first axis2_placement_3d in transformation is SELF and second is from parent>	PATH		103	axis2_placement_3d <= placement <= geometric_representation_item <= representation_item <= item_defined_transformation.transform_item_1 item_defined_transformation {item_defined_transformation.name = 'transformation to parent'} item_defined_transformation.transform_item_2 -> representation_item => geometric_representation_item => placement => axis2_placement_3d {CLASS_HELP<axis2_placement_3d> (group.name = 'local co-ordinate system') (group.name = 'local co-ordinate system with position reference') (group.name = 'global axis placement') class}
-from DEFINITION				
local_co_ordinate_system to global_id (as id)	extended_property_definition.id	218	99, 104	
local_co_ordinate_system to unit (as local_units) <SET>	PATH			extended_property_definition <= PROP_DEF_TO_UNITS<'local co-ordinate system'>
local_co_ordinate_system to definable_object (as defined_for) <SET[1:?]> #1: defined for a part, system #2: defined for a feature	#1: PATH #2: PATH			#1: (extended_property_definition <= PROP_TO_PROD_DEF) #2: (extended_property_definition <= property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect

Application Element	AIM element	Source	Rules	Reference Path
				shape_aspect => extended_shape_aspect)
local_co_ordinate_system to definable_object (as defined_for) <SET[1:?]>	PATH			extended_property_definition <= PROP_TO_PROD_DEF
-from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	98	VERSION_ID<extended_property_definition>
-from AXIS2_PLACEMENT_3D				
local_co_ordinate_system to direction (as axis) <OPTIONAL>	PATH			axis2_placement_3d axis2_placement_3d.axis -> direction
local_co_ordinate_system to direction (as ref_direction) <OPTIONAL>	PATH			axis2_placement_3d axis2_placement_3d.ref_direction -> direction
-from PLACEMENT				
local_co_ordinate_system to cartesian_point (as location)	PATH			axis2_placement_3d <= placement placement.location -> cartesian_point
-from REPRESENTATION_ITEM				
name	property_definition.name	41		
LOCAL_CO_ORDINATE_SYSTEM_WITH_POSITION_REFERENCE	[extended_property_definition] [axis2_placement_3d]	218 42	106, 107, 97	{[CLASS<extended_property_definition, 'local co-ordinate system with position reference', 'local co-ordinate system'>] [CLASS<extended_property_definition, 'local co-ordinate system', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]] PROP_DEF_REP_HELP<'local co-ordinate system with position reference'> representation {representation.name = 'local axis with position reference representation' representation.context_of_items representation_context =>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {representation_context.context_type = 'local co-ordinate space'} geometric_representation_context {geometric_representation_context.coordinate_space_dimension = 3} REP_ITEM<'local axes and origin'> geometric_representation_item => placement => axis2_placement_3d </pre>
local_co_ordinate_system_with_position_reference to cartesian_point (as location) <DERIVE>	PATH			<pre> axis2_placement_3d <= placement placement.location -> cartesian_point </pre>
longitudinal_ref <OPTIONAL>	(value_representation_item.value_component) (compound_representation_item)	43 43		<pre> extended_property_definition<= PROP_DEF_REP_HELP<'local co-ordinate system with position reference'> representation {representation.name = 'local axis with position reference representation'} representation.items [i] -> representation_item (representation_item => compound_representation_item {CLASS_HELP<compound_representation_item> (group.name = 'longitudinal position') (group.name = 'spacing position with offset') class}) (representation_item => {representation_item.name = 'longitudinal ref'} value_representation_item {value_representation_item.value_component -> measure_value = length_measure}) </pre>
transversal_ref <OPTIONAL>	(value_representation_item.value_component) (compound_representation_item)	43 43		<pre> extended_property_definition<= PROP_DEF_REP_HELP<'local co-ordinate system with position reference'> representation {representation.name = 'local axis with position reference </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> representation' } representation.items [i] -> representation_item (representation_item => compound_representation_item { CLASS_HELP<compound_representation_item> (group.name = 'transversal position') (group.name = 'spacing position with offset') class }) (representation_item => { representation_item.name = 'transversal ref' } value_representation_item { value_representation_item.value_component -> measure_value = length_measure }) </pre>
vertical_ref <OPTIONAL>	<pre> (value_representation_item.value_component) (compound_representation_item) </pre>	43 43		<pre> extended_property_definition<= PROP_DEF_REP_HELP<'local co-ordinate system with position reference'> representation { representation.name = 'local axis with position reference representation' } representation.items [i] -> representation_item (representation_item => compound_representation_item { CLASS_HELP<compound_representation_item> (group.name = 'vertical position') (group.name = 'spacing position with offset') class }) (representation_item => { representation_item.name = 'vertical ref' } value_representation_item { value_representation_item.value_component -> measure_value = length_measure }) </pre>
-from LOCAL_CO_ORDINATE_SYSTEM				
parent				<i>see supertype mappings</i>
-from DEFINITION				

Application Element	AIM element	Source	Rules	Reference Path
local_co_ordinate_system_with_position_reference to global_id (as id)				<i>see supertype mappings</i>
local_co_ordinate_system_with_position_reference to unit (as local_units) <SET>				<i>see supertype mappings</i>
local_co_ordinate_system_with_position_reference to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
-from AXIS2_PLACEMENT_3D				
local_co_ordinate_system_with_position_reference to direction (as axis) <OPTIONAL>				<i>see supertype mappings</i>
local_co_ordinate_system_with_position_reference to direction (as ref_direction) <OPTIONAL>				<i>see supertype mappings</i>
-from REPRESENTATION_ITEM				
name				<i>see supertype mappings</i>
SPACING_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, 'spacing table', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]}
description <OPTIONAL>	property_definition.description	43		
name <OPTIONAL>	property_definition.name	43		
spacing_table to spacing_position (as spacing_table_representations) <LIST[0:?]>	PATH			extended_property_definition<= PROP_DEF_TO_REP representation.items [i] -> representation_item => compound_representation_item {CLASS_HELP<compound_representation_item>

Application Element	AIM element	Source	Rules	Reference Path
				(group.name = 'longitudinal position') (group.name = 'transversal position') (group.name = 'vertical position') (group.name = 'spacing position with offset') class}
-from DEFINITION				
spacing_table to global_id (as id)	extended_property_definition.id	218	99, 105	
spacing_table to unit (as local_units) <SET>	PATH			extended_property_definition <= PROP_DEF_TO_UNITS<'spacing table'>
spacing_table to definable_object (as defined_for) <SET[1:?]>	PATH			extended_property_definition <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition (characterized_product_definition = product_definition) (characterized_product_definition = product_definition product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product)
-from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	98	VERSION_ID<extended_property_definition>
LONGITUDINAL_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, 'longitudinal table', 'spacing table'>] [CLASS<extended_property_definition, 'spacing table', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]}
longitudinal_table to longitudinal_position (as spacing_table_representations) <LIST[0:?]>	PATH			extended_property_definition<= PROP_DEF_TO_REP representation.items [i] -> representation_item =>

Application Element	AIM element	Source	Rules	Reference Path
				compound_representation_item {CLASS_ID<compound_representation_item, 'longitudinal position'>}
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
longitudinal_table to global_id (as id)				<i>see supertype mappings</i>
longitudinal_table to unit (as local_units) <SET>				<i>see supertype mappings</i>
longitudinal_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
STATION_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, 'station table', 'longitudinal table'>] [CLASS<extended_property_definition, 'longitudinal table', 'spacing table'>] [CLASS<extended_property_definition, 'spacing table', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]}
-from LONGITUDINAL_TABLE				
station_table to longitudinal_position (as spacing_table_representations) <LIST[0:?]>				<i>see supertype mappings</i>
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
station_table to global_id (as id)				<i>see supertype mappings</i>
station_table to unit				<i>see supertype mappings</i>

Application Element	AIM element	Source	Rules	Reference Path
(as local_units) <SET>				
station_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
FRAME_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, ‘frame table’, ‘longitudinal table’>] [CLASS<extended_property_definition, ‘longitudinal table’, ‘spacing table’> [CLASS<extended_property_definition, ‘spacing table’, ‘definition’> [CLASS<extended_property_definition, ‘definition’, ‘versionable object’>] ROOT_CLASS<extended_property_definition, ‘versionable object’>]}
-from LONGITUDINAL_TABLE				
frame_table to longitudinal_position (as spacing_table_representations) <LIST[0:?]>				<i>see supertype mappings</i>
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
frame_table to global_id (as id)				<i>see supertype mappings</i>
frame_table to unit (as local_units) <SET>				<i>see supertype mappings</i>
frame_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
TRANSVERSAL_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, ‘transversal table’, ‘spacing table’> [CLASS<extended_property_definition, ‘spacing table’, ‘definition’> [CLASS<extended property definition, ‘definition’, ‘versionable

Application Element	AIM element	Source	Rules	Reference Path
				object’>] ROOT_CLASS<extended_property_definition, ‘versionable object’>]]
transversal_table to transversal_position (as spacing_table_representations) <LIST[0:?]>	PATH			extended_property_definition<= PROP_DEF_TO_REP representation.items [i] -> representation_item => compound_representation_item {CLASS_ID<compound_representation_item, ‘transversal position’>}
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
transversal_table to global_id (as id)				<i>see supertype mappings</i>
transversal_table to unit (as local_units) <SET>				<i>see supertype mappings</i>
transversal_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
BUTTOCK_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, ‘buttock table’, ‘transversal table’>] [CLASS<extended_property_definition, ‘transversal table’, ‘spacing table’>] [CLASS<extended_property_definition, ‘spacing table’, ‘definition’>] [CLASS<extended_property_definition, ‘definition’, ‘versionable object’>] ROOT_CLASS<extended_property_definition, ‘versionable object’>]]
-from TRANSVERSAL_TABLE				
buttock_table to transversal_position (as spacing_table_representations) <LIST[0:?]>				<i>see supertype mappings</i>

Application Element	AIM element	Source	Rules	Reference Path
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
buttock_table to global_id (as id)				<i>see supertype mappings</i>
buttock_table to unit (as local_units) <SET>				<i>see supertype mappings</i>
buttock_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
VERTICAL_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, 'vertical table', 'spacing table'>] [CLASS<extended_property_definition, 'spacing table', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]}
vertical_table to vertical_position (as spacing_table_representations) <LIST[0:?]	PATH			extended_property_definition<= PROP_DEF_TO_REP representation.items [i] -> representation_item => compound_representation_item { CLASS_ID<compound_representation_item, 'vertical position'> }
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
vertical_table to global_id (as id)				<i>see supertype mappings</i>
vertical_table to unit (as local_units) <SET>				<i>see supertype mappings</i>

Application Element	AIM element	Source	Rules	Reference Path
vertical_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
WATERLINE_TABLE	extended_property_definition	218		{[CLASS<extended_property_definition, 'waterline table', 'vertical table'>] [CLASS<extended_property_definition, 'vertical table', 'spacing table'>] [CLASS<extended_property_definition, 'spacing table', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] ROOT_CLASS<extended_property_definition, 'versionable object'>]}
-from VERTICAL_TABLE				
vertical_table to vertical_position (as spacing_table_representations) <LIST[0:?]>				<i>see supertype mappings</i>
-from SPACING_TABLE				
description <OPTIONAL>				<i>see supertype mappings</i>
name <OPTIONAL>				<i>see supertype mappings</i>
-from DEFINITION				
vertical_table to global_id (as id)				<i>see supertype mappings</i>
vertical_table to unit (as local_units) <SET>				<i>see supertype mappings</i>
vertical_table to definable_object (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
-from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
SPACING_POSITION	compound_representation_item	43		{ROOT_CLASS<compound_representation_item, 'spacing position'>} {compound_representation_item <- representation.items[i] representation representation.context_of_items

Application Element	AIM element	Source	Rules	Reference Path
				representation_context => {representation_context.context_type = 'global co-ordinate space'} geometric_representation_context}
name <OPTIONAL>	compound_representation_item .name	43		
position_number	value_representation_item.value_component	43	108, 97	COMPOUND<'position number'> value_representation_item {value_representation_item.value_component -> measure_value = count_measure}
position	value_representation_item.value_component	43	108, 97	COMPOUND<'position'> value_representation_item {value_representation_item.value_component -> measure_value = length_measure}
LONGITUDINAL_POSITION	compound_representation_item	43		{CLASS<compound_representation_item, 'longitudinal position', 'spacing position'> ROOT_CLASS<compound_representation_item, 'spacing position'>} {compound_representation_item <- representation.items[i] representation representation.context_of_items representation_context => {representation_context.context_type = 'global co-ordinate space'} geometric_representation_context}
-from SPACING_POSITION				
name <OPTIONAL>				<i>see supertype mappings</i>
position_number				<i>see supertype mappings</i>
position				<i>see supertype mappings</i>
TRANSVERSAL_POSITION	compound_representation_item	43		{CLASS<compound_representation_item, 'transversal position', 'spacing position'> ROOT_CLASS<compound_representation_item, 'spacing position'>} {compound_representation_item <- representation.items[i]

Application Element	AIM element	Source	Rules	Reference Path
				representation representation.context_of_items representation_context => {representation_context.context_type = 'global co-ordinate space'} geometric_representation_context}
-from SPACING_POSITION				
name <OPTIONAL>				<i>see supertype mappings</i>
position_number				<i>see supertype mappings</i>
position				<i>see supertype mappings</i>
VERTICAL_POSITION	compound_representation_item	43		{CLASS<compound_representation_item, 'vertical position', 'spacing position'> ROOT_CLASS<compound_representation_item, 'spacing position'>} {compound_representation_item <- representation.items[i] representation representation.context_of_items representation_context => {representation_context.context_type = 'global co-ordinate space'} geometric_representation_context}
-from SPACING_POSITION				
name <OPTIONAL>				<i>see supertype mappings</i>
position_number				<i>see supertype mappings</i>
position				<i>see supertype mappings</i>
SPACING_POSITION_WITH_OFFSET	compound_representation_item	43		{CLASS<compound_representation_item, 'spacing position with offset', 'spacing position'> ROOT_CLASS<compound_representation_item, 'spacing position'>} {compound_representation_item <- representation.items[i] representation representation.context_of_items representation_context => {representation_context.context_type = 'global co-ordinate

Application Element	AIM element	Source	Rules	Reference Path
				space'} geometric_representation_context}
offset	value_representation_item.value_component	43	109, 97	COMPOUND<'offset'> value_representation_item {value_representation_item.value_component -> measure_value = length_measure}
spacing_position_reference to spacing_position (as relating_spacing_position)	PATH		110	COMPOUND<'relating spacing position'> {CLASS_HELP<compound_representation_item> (group.name = 'longitudinal position') (group.name = 'transversal position') (group.name = 'vertical position') class}
position <DERIVE>	value_representation_item.value_component	43		COMPOUND<'position'> value_representation_item {value_representation_item.value_component -> measure_value = length_measure}
-from SPACING_POSITION				
name <OPTIONAL>				<i>see supertype mappings</i>
position_number				<i>see supertype mappings</i>

Table 9 - Mapping table for product_structures UoF

Application Element	AIM element	Source	Rules	Reference Path
ASSEMBLY	[product_definition] [group]	41		{[LINK_TO_GROUP<product_definition>] [CLASS<product_definition, 'assembly', 'item structure'>] [CLASS<product_definition, 'item structure', 'definable object'>] [CLASS<product_definition, 'item structure', 'versionable object'>] [ROOT_CLASS<product_definition, 'versionable object'>] [CLASS<product_definition, 'assembly', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>] [(CLASS<product_definition, 'unit assembly', 'assembly'>] (CLASS<product_definition, 'production panel', 'assembly'>) (CLASS<product_definition, 'preconstructed group', 'assembly'>) (CLASS<product_definition, 'preconstructed section', 'assembly'>) (CLASS<product_definition, 'user defined', 'assembly'>)]}
the_class	group.name	41		product_definition = classification_item <- applied_classification_assignment.items[i] applied_classification_assignment applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'class membership'}

Application Element	AIM element	Source	Rules	Reference Path
				classification_assignment.assigned_classification -> group group.name {(group.name = 'unit assembly') (group.name = 'production panel') (group.name = 'preconstructed group') (group.name = 'preconstructed section')}
- from ITEM				
description	product_definition.description	41		
name	product_definition.name	41		
assembly to external_reference (as documentation) (S[1:?])	PATH			product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = '.UNUSED. ']}
assembly to ship (as ship_context)	PATH			product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structures'} product_definition_formation.of_product -> product {CLASS_ID<product, 'ship'>}
- from ITEM_STRUCTURE				
assembly to external_instance_reference (as external_items) (OPT)	PATH			group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'part'>] [EXT_INST_REF<product_definition, 'ship structures', 'part'>])}

Application Element	AIM element	Source	Rules	Reference Path
				<pre> ([CLASS_ID<product_definition, 'assembly'>] [EXT_INST_REF<product_definition, 'ship structures', 'assembly'>]]) </pre>
assembly to external_instance_reference (as external_relationships) (OPT)	PATH			<pre> group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition_relationship, 'part relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'part relationship'>]) ([CLASS_ID<product_definition_relationship, 'assembly relationship'>] [EXT_INST_REF<product_definition, 'ship structures', 'assembly relationship'>])} </pre>
assembly to item (as items) (OPT)	PATH			<pre> group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'part'>] ([CLASS_ID<product_definition, 'assembly'>])} </pre>
assembly to item_relationship (as relationships) (OPT)	PATH			<pre> group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition_relationship, 'part relationship'>] ([CLASS_ID<product_definition_relationship, 'assembly relationship'>])} </pre>
- from DEFINABLE_OBJECT				
assembly to definition (as definitions) (INV)	PATH			<pre> product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition </pre>
assembly to global_id (as id)	product_definition.id	41	111, 112	

Application Element	AIM element	Source	Rules	Reference Path
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<product_definition>
ASSEMBLY_RELATIONSHIP	product_definition_relations hip	41	124	{[CLASS<product_definition_relationship, 'assembly_relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]}
assembly_relationship to assembly (as item_1)	product_definition_relations hip.relati ng_ product_definition	41		product_definition_relationship.related_product_ definition -> product_definition {CLASS_ID<product_definition, 'assembly'>}
- from ITEM_RELATIONSHIP contxt	product_definition.descripti on	41		
assembly_relationship to external_instance_reference (as external_item_1) (OPT)	product_definition_relations hip.relati ng_ product_definition	41		{product_definition_relationship.relati ng_product_ definition -> product_definition [CLASS_ID<product_definition, 'assembly'>] [EXT_INST_REF<product_definition, 'ship structures', 'assembly'>]]}
assembly_relationship to external_instance_reference (as external_item_2) (OPT)	product_definition_relations hip.related_ product_definition	41		{product_definition_relationship.related_product_ definition -> product_definition [CLASS_ID<product_definition, 'assembly'>] [EXT_INST_REF<product_definition, 'ship structures',

Application Element	AIM element	Source	Rules	Reference Path
				'assembly'>]]
assembly_relationship to item (as item_2) (OPT)	product_definition_relations hip.related_ product_definition	41		{CLASS_ID<product_definition, 'assembly'>}
- from DEFINABLE_OBJECT				
assembly_relationship to definition (as definitions) (INV)	PATH			product_definition_relationship = characterized_product_definition = characterized_definition <- property_definition.definition property_definition
assembly_relationship to global_id (as id)	product_definition_relations hip.id	41	111, 113	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<product_definition_relationship>
COMPOSITE_FEATURE	extended_shape_aspect	218	116	{[CLASS<shape_aspect, 'composite feature', 'feature'>] [CLASS<shape_aspect, 'feature', 'item'>] [CLASS<shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<shape_aspect, 'definable object'>]}
composite_feature to feature (as composed_of)	PATH		118, 122	extended_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatenship_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'composed of' } shape_aspect_relationship.related_shape_aspect -> shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'corner cutout'>) (CLASS_ID<extended_shape_aspect, 'edge cutout'>) (CLASS_ID<extended_shape_aspect, 'interior cutout'>) (CLASS_ID<extended_shape_aspect, 'edge bevel'>) (CLASS_ID<extended_shape_aspect, 'seam'>)}

Application Element	AIM element	Source	Rules	Reference Path
- from ITEM				
description	shape_aspect.description	41		extended_shape_aspect <= shape_aspect shape_aspect.description
name	shape_aspect.name	41		extended_shape_aspect <= shape_aspect shape_aspect.name
composite_feature to external_reference (as documentation) (S[1:?])	PATH			<i>See supertype</i>
composite_feature to ship (as ship_context)	PATH			<i>See supertype</i>
- from DEFINABLE_OBJECT				
composite_feature to definition (as definitions) (INV)	PATH			<i>See supertype</i>
composite_feature to global_id (as id)	extended_shape_aspect.id	218	111, 115	
EXPLICIT_FEATURE_DESIGN_- DEFINITION	representation	43	118, 120	representation {[CLASS<representation, 'explicit feature design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]}
explicit_feature_design_definition to shape_representation (as representations) (S[1:?])	PATH		121	representation REP2SA_DN<'feature parameter representation', 'representations' shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'corner cutout'>) (CLASS_ID<extended_shape_aspect, 'edge feature'>)} {(CLASS_ID<extended_shape_aspect, 'corner cutout'>) (CLASS_ID<extended_shape_aspect, 'edge feature'>)}

Application Element	AIM element	Source	Rules	Reference Path
				{CLASS_ID<extended_shape_aspect, 'interior cutout'>} {CLASS_ID<extended_shape_aspect, 'edge bevel'>} {CLASS_ID<extended_shape_aspect, 'seam'>} <= shape_aspect SA2REP_D<'feature shape representation' representation {representation.name = 'representations'} => shape_representation
- from FEATURE_DESIGN_DEFINITION				
explicit_feature_design_definition to feature (as defined_for) (S[1:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'corner cutout'>} {CLASS_ID<extended_shape_aspect, 'edge cutout'>} {CLASS_ID<extended_shape_aspect, 'interior cutout'>} {CLASS_ID<extended_shape_aspect, 'edge bevel'>} {CLASS_ID<extended_shape_aspect, 'seam'>}
- from DEFINITION				
explicit_feature_design_definition to global_id (as id)	representation.id	41	111, 119	
explicit_feature_design_definition to unit (as local_units)) (S[1:?])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			
FEATURE (ABS)	extended_shape_aspect	218		<i>See supertype</i>
- from ITEM				
description	shape_aspect.description	41		{extended_shape_aspect <= shape_aspect shape_aspect.description}
name	shape_aspect.name	41		{extended_shape_aspect <= shape_aspect

Application Element	AIM element	Source	Rules	Reference Path
feature to external_reference (as documentation)) (S[1:?])	PATH			shape_aspect.name } extended_shape_aspect <= shape_aspect <= applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = 'UNUSED. ']}
feature to ship (as ship_context)	PATH			extended_shape_aspect <= shape_aspect = shape_definition = characterized_definition = characterized_product_definition = product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structures'} product_definition_formation.of_product -> product {CLASS_ID<product, 'ship'>}
- from DEFINABLE_OBJECT				
feature to definition (as definitions) (INV)	PATH			extended_shape_aspect <= shape_aspect = represented_definition <= PDR -> representation
feature to global_id (as id)	extended_shape_aspect.id	218	111, 115	
FEATURE DESIGN DEFINITION (ABS)	representation	43		<i>See subtypes</i>
- from DESIGN DEFINITION				

Application Element	AIM element	Source	Rules	Reference Path
feature_design_definition to shape_representation (as representations)	<i>See subtypes</i>			
- from DEFINITION				
feature_design_definition to global_id (as id)	representation.id	41	111, 119	
feature_design_definition to unit (as local_units) (S[1:?])	PATH			<pre> representation representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']} representation_context => global_unit_assigned_context global_unit_assigned_context.units -> unit </pre>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<representation>
FEATURE_RELATIONSHIP (ABS)	extended_shape_aspect_rela tionship	218		<i>See subtypes</i>
feature_relationship to feature (as item_1) (OPT)	shape_aspect_relationship. relating_shape_aspect	41		<i>See subtypes</i>
- from ITEM_RELATIONSHIP contxt	shape_aspect_relationship. description	41		<pre> extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.description </pre>
feature_relationship to external_instance_reference (as external_item_1) (OPT)	shape_aspect_relationship. relating_shape_aspect	41		<i>See subtypes</i>

Application Element	AIM element	Source	Rules	Reference Path
feature_relationship to external_instance_reference (as external_item_2) (OPT)	shape_aspect_relationship. related_shape_aspect	41		<i>See subtypes</i>
feature_relationship to item (as item_2) (OPT)	shape_aspect_relationship. related_shape_aspect	41		<i>See subtypes</i>
- from DEFINABLE_OBJECT				
feature_relationship to definition (as definitions) (INV)	PATH			<i>See subtypes</i>
feature_relationship to global_id (as id)	extended_shape_aspect_rela tionship.id	218	111, 117	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<extended_shape_aspect_relationship>
PART (ABS)	product_definition	41		<i>See subtypes</i>
- from ITEM				
name	product_definition.name	41		
description	product_definition.descripti on	41		
part to external_reference (as documentation)) (SET)	PATH			<pre> product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = 'UNUSED. ']} </pre>
part to ship (as ship_context)	PATH			<pre> product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structure'} product_definition_formation.of_product -> </pre>

Application Element	AIM element	Source	Rules	Reference Path
				product {CLASS_ID<product, 'ship'>}
- from DEFINABLE_OBJECT				
part to definition (as definitions) (INV)	PATH			product_definition = characterized_product_definition = characterized_definition <- property_definition.definition (property_definition) (property_definition => product_definition_shape)
part to global_id (as id)	product_definition.id	41	111, 112	
PART_RELATIONSHIP (ABS)	product_definition_relations hip	41		<i>See subtypes</i>
part_relationship to part (as item_1) (OPT)	product_definition_relations hip.relatng_- product_definition	41		<i>See subtypes</i>
- from ITEM_RELATIONSHIP				
contxt	product_definition_relations hip.description	41		
part_relationship to external_instance_reference (as external_item_1) (OPT)	product_definition_relations hip.relatng_- product_definition	41		<i>See subtypes</i>
part_relationship to external_instance_reference (as external_item_2) (OPT)	product_definition_relations hip.related_- product_definition	41		<i>See subtypes</i>
part_relationship to item (as item_2) (OPT)	product_definition_relations hip.related_- product_definition	41		<i>See subtypes</i>
- from DEFINABLE_OBJECT				
part_relationship to definition (as definitions) (INV)	PATH			product_definition_relationship = characterized_product_definition = characterized_definition <-

Application Element	AIM element	Source	Rules	Reference Path
				property_definition.definition property_definition
part_relationship to global_id (as id)	product_definition_relations hip.id	41	111, 113	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<product_definition_relationship>
SYSTEM (ABS)	[product_definition] [group]	41 41		<i>See subtypes</i>
- from ITEM				
description	product_definition.descripti on	41		
name	product_definition.name	41		
system to external_reference (as documentation) (S[1:?])	PATH			product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = '.UNUSED. ']} }
system to ship (as ship_context)	PATH			product_definition product_definition.formation -> product_definition_formation { product_definition_formation.id = 'ship structures' product_definition_formation.of_product -> product { CLASS_ID<product, 'ship'> } }
- from ITEM_STRUCTURE				

Application Element	AIM element	Source	Rules	Reference Path
system to external_instance_reference (as external_items) (S[0:?])	PATH			group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'structural system'>] [EXT_INST_REF<product_definition, 'ship structures', 'structural system'>]) ([CLASS_ID<product_definition, 'system'>] [EXT_INST_REF<product_definition, 'ship structures', 'system'>])])}
system to external_instance_reference (as external_relationships) (S[0:?])	PATH			group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition_relationship, 'structural system relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural system relationship'>]) ([CLASS_ID<product_definition_relationship, 'system relationship'>] [EXT_INST_REF<product_definition, 'ship structures', 'system relationship'>])])}
system to item (as items) (S[0:?])	PATH			group <- GROUPS<product_definition, 'item structure'> {(CLASS_ID<product_definition, 'structural system'> (CLASS_ID<product_definition, 'system'>))}
system to item_relationship (as relationships) (S[0:?])	PATH			group <- GROUPS<product_definition_relationship, 'item structure'> {(CLASS_ID<product_definition_relationship, 'structural system relationship'>) (CLASS_ID<product_definition_relationship, 'system relationship'>))}
- from DEFINABLE_OBJECT				
system to definition (as definitions) (INV)	PATH			product_definition = characterized_product_definition = characterized_definition <-

Application Element	AIM element	Source	Rules	Reference Path
				property_definition.definition property_definition
system to global_id (as id)	product_definition.id	41	111, 112	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<product_definition>
SYSTEM_DESIGN_DEFINITION (ABS)	extended_product_definition _shape	218		<i>See subtypes</i>
system_design_definition to system (as defined_for) (S[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'system'>}
- from DEFINITION				
system_design_definition to global_id (as id)	extended_product_definitio n_shape.id	218	111, 123	
system_design_definition to unit (as local_units) (S[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition {property_definition.name = '.UNUSED:'} property_definition <- represented_definition <- PDR -> representation representation.context_of_items ->

Application Element	AIM element	Source	Rules	Reference Path
				<pre> representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']} representation_context => global_unit_assigned_context global_unit_assigned_context.units -> unit </pre>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<extended_product_definition_shape>
SYSTEM_RELATIONSHIP (ABS)	product_definition_relations hip	41		<i>See subtypes</i>
system_relationship to system (as item_1) (OPT)	product_definition_relations hip.relatig_- product_definition	41		<i>See subtypes</i>
- from ITEM_RELATIONSHIP				
contxt	product_definition_relations hip.description	41		
system_relationship to external_instance_reference (as external_item_1) (OPT)	product_definition_relations hip.relatig_- product_definition	41		<i>See subtypes</i>
system_relationship to external_instance_reference (as external_item_2) (OPT)	product_definition_relations hip.related_- product_definition	41		<i>See subtypes</i>
system_relationship to item (as item_2) (OPT)	product_definition_relations hip.related_- product_definition	41		<i>See subtypes</i>
- from DEFINABLE_OBJECT				
system_relationship to definition (as definitions) (INV)	PATH			<pre> product_definition_relationship = characterized_product_definition = characterized_definition <- property_definition.definition </pre>

Application Element	AIM element	Source	Rules	Reference Path
				property_definition
system_relationship to global_id (as id)	product_definition_relations hip.id	41	111, 113	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	114	VERSION_ID<product_definition_relationship>

Table 10 - Mapping table for ship_general_characteristics UoF

Application Element	AIM Element	Source	Rules	Reference Path
PRINCIPAL_CHARACTERISTICS	product_definition	41	125	{[CLASS<product_definition, 'principal characteristics', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
length_between_perpendiculars	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'length between perpendiculars', positive_length_measure>
moulded_breadth	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'moulded breadth', positive_length_measure>

Application Element	AIM Element	Source	Rules	Reference Path
moulded_depth	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'moulded depth', positive_length_measure>
design_draught	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'design draught', positive_length_measure>
block_coefficient <OPTIONAL>	value_representation_item.value_component	43	128, 127	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'block coefficient', ratio_measure>
design_deadweight	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'design deadweight', mass_measure>
min_draught_at_fp	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'min draught at fp', positive_length_measure>
max_draught_at_fp	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'max draught at fp', positive_length_measure>

Application Element	AIM Element	Source	Rules	Reference Path
min draught at ap	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'min draught at ap', positive_length_measure>
max draught at ap	value_representation_item.value_component	43	128, 126	PROD_DEF_TO_VAL_REP_ITEM<'principal characteristics', 'max draught at ap', positive_length_measure>
- from GENERAL_CHARACTERISTICS_DEFINITION				
principal_characteristics to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
principal_characteristics to global_id (as id)	product_definition.id	41	130, 131	
principal_characteristics to unit (as local_units) <SET>	PATH			PROD_DEF_TO_UNITS<'principal characteristics'>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<product_definition>
CLASS_AND_STATUTORY_DESIGNATION	product_definition	41	137	{[CLASS<product_definition, 'class and statutory designation', 'general characteristics definition'>]}

Application Element	AIM Element	Source	Rules	Reference Path
				[CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
class_number	descriptive_representation_item.description	45	138	PROD_DEF_TO_DESC_REP_ITEM<'class and statutory designation', 'class number'>
class_and_statutory_designation to class_notation (as the_class)	PATH		139	PROD_DEF_PROP_DEF {CLASS_ID<property_definition, 'class notation'>}
class_and_statutory_designation to regulation (as the_statutory)	PATH		140	PROD_DEF_PROP_DEF {CLASS_ID<property_definition, 'regulation'>}
- from GENERAL_CHARACTERISTICS_DEFINITION				
class_and_statutory_designation to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
class_and_statutory_designation to global_id (as id)	product_definition.id	41	130, 132	
class_and_statutory_designation to unit (as local_units) <SET[0:0]> <no mapping required>				
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<product_definition>

Application Element	AIM Element	Source	Rules	Reference Path
CLASS_NOTATION	property_definition	41	141	{CLASS_ID<property_definition, 'class notation'>}
class_notation to organisation (as class_society)	PATH		142	ORG_ASSGN<property_definition, 'class society'>
class_notations_hull <SET[1:?]>	descriptive_representation_it em.description	45	143	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'class notations hull'>
class_notations_machinery <SET[1:?]>	descriptive_representation_it em.description	45	143	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'class notations machinery'>
ice_class_notation <OPTIONAL>	descriptive_representation_it em.description	45	144	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'ice class notation'>
service_area	descriptive_representation_it em.description	45	145	PROP_DEF_TO_DESC_REP_ITEM<'class society', 'service area'>

Application Element	AIM Element	Source	Rules	Reference Path
service_factor <OPTIONAL>	value_representation_item.value_component	43	128, 144	PROP_DEF_TO_VAL_REP_ITEM<'class notation', 'service factor', count_measure>
approval_required_for_oil_cargo	descriptive_representation_item.description	45	145	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'approval required for oil cargo'> {(descriptive_representation_item.description = 'TRUE') (descriptive_representation_item.description = 'FALSE')}
approval_required_loading_unloading_aground	descriptive_representation_item.description	45	145	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'approval required for loading unloading aground'> {(descriptive_representation_item.description = 'TRUE') (descriptive_representation_item.description = 'FALSE')}
approval_required_loading_unloading_grabs	descriptive_representation_item.description	45	145	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'approval required for unloading grabs'> {(descriptive_representation_item.description = 'TRUE') (descriptive_representation_item.description = 'FALSE')}

Application Element	AIM Element	Source	Rules	Reference Path
approval_required_for_heavy_cargo <OPTIONAL>	descriptive_representation_item.description	45	144	PROP_DEF_TO_DESC_REP_ITEM<'class notation', 'approval required for heavy cargo'> {(descriptive_representation_item.description = 'HC') (descriptive_representation_item.description = 'HC_E') (descriptive_representation_item.description = 'HC_EA')}
REGULATION	property_definition	41		{CLASS_ID<property_definition, 'regulation'>}
regulation to document_reference (as international_regulations) <SET[0:?]>	PATH	41		DOC_REF<property_definition, 'international regulations'>
regulation to document_reference (as standards) <SET[0:?]>	PATH	41		DOC_REF<property_definition, 'standards'>
regulation to document_reference (as national_regulations) <SET[0:?]>	PATH	41		DOC_REF<property_definition, 'national regulations'>
CLASS_PARAMETERS	product_definition	41	146	{[CLASS<product_definition, 'class parameters', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>]}

Application Element	AIM Element	Source	Rules	Reference Path
				[CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
length_class	value_representation_item.value_component	43	128, 147	PROD_DEF_TO_VAL_REP_ITEM<'class parameters', 'length class', positive_length_measure>
length_solas	value_representation_item.value_component	43	128, 147	PROD_DEF_TO_VAL_REP_ITEM<'class parameters', 'length solas', positive_length_measure>
scantlings draught	value_representation_item.value_component	43	128, 147	PROD_DEF_TO_VAL_REP_ITEM<'class parameters', 'scantlings draught', positive_length_measure>
block_coefficient_class	value_representation_item.value_component	43	128, 147	PROD_DEF_TO_VAL_REP_ITEM<'class parameters', 'block coefficient class', ratio_measure>

Application Element	AIM Element	Source	Rules	Reference Path
design_speed_ahead	value_representation_item.value_component	43	128, 147	PROD_DEF_TO_SPECIAL_VAL_REP_ITEM<'class parameters', 'design speed ahead', speed_unit>
design_speed_astern	value_representation_item.value_component	43	128, 147	PROD_DEF_TO_SPECIAL_VAL_REP_ITEM<'class parameters', 'design speed astern', speed_unit>
- from GENERAL_CHARACTERISTICS_DEFINITION				
class_parameters to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
class_parameters to global_id (as id)	product_definition.id	41	130, 131	
class_parameters to unit (as local_units) <SET>	PATH			PROD_DEF_TO_UNITS<'class parameters'>
- from VERSIONABLE_OBJECT				

Application Element	AIM Element	Source	Rules	Reference Path
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<product_definition>
SHIP_DESIGNATION	product_definition	41		{[CLASS<product_definition, 'ship designation', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
ship_name	product_definition.name	41		
ship_type_description <should be deleted ARM issue>	product_definition.description	41		
ship_identification	identification_assignment.assigned_id	41	148	HAS_ID_2_ROLES<product_definition, 'IMO number', 'pennant hull number'>
call_sign	identification_assignment.assigned_id	41	149	HAS_ID_1_ROLE<product_definition, 'call sign'>
flag_state	identification_assignment.assigned_id	41	150	HAS_ID_1_ROLE<product_definition, 'flag state'>
port_of_registration	identification_assignment.assigned_id	41	151	HAS_ID_1_ROLE<product_definition, 'port of registration'>
- from GENERAL_CHARACTERISTICS_DEFINITION				

Application Element	AIM Element	Source	Rules	Reference Path
N				
ship_designation to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
ship_designation to global_id (as id)	product_definition.id	41	130, 133	
ship_designation to unit (as local_units) <SET[0:0]> <no mapping required>				
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<product_definition>
SHIPYARD_DESIGNATION	product_definition	41		{[CLASS<product_definition, 'shipyard designation', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
role	organization_assignment.role	41		ORG_ASS_PART<product_definition> { organization_assignment.role -> organization_role {(organization_role.name = 'prime design') (organization_role.name = 'prime build') (organization_role.name = 'prime repair') (organization_role.name = 'prime') (organization_role.name = 'subcontractor')}}}

Application Element	AIM Element	Source	Rules	Reference Path
shipyard_new_building_id <OPTIONAL>	identification_assignment.assigned_id	41		HAS_ID_1_ROLE<product_definition, 'shipyard new building id'>
shipyard_project_name	organizational_project.name	41		ORG_ASSGN<product_definition, 'shipyard'> <- organizational_project.responsible_organizations[i] organizational_project {organizational_project.description = 'shipyard project name'} organizational_project organizational_project.name
shipyard_designation to organisation (as shipyard_name)	organization_assignment.assigned_organization	41	152	ORG_ASSGN<product_definition, 'shipyard'>
- from GENERAL_CHARACTERISTICS_DEFINITION				
shipyard_designation to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
shipyard_designation to global_id (as id)	product_definition.id	41	130, 134	
shipyard_designation to unit (as local_units) <SET[0:0]> <no mapping required>				
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<product_definition>

Application Element	AIM Element	Source	Rules	Reference Path
OWNER_DESIGNATION	product_definition	41		{[CLASS<product_definition, 'owner designation', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
owner_designation to organisation (as managing_company)	PATH		153	ORG_ASSGN<product_definition, 'managing company'>
owner_designation to organisation (as ordering_company)	PATH		154	ORG_ASSGN<product_definition, 'ordering company'>
owner_designation to organisation (as owning_company)	PATH		155	ORG_ASSGN<product_definition, 'owning company'>
- from GENERAL_CHARACTERISTICS_DEFINITION				
owner_designation to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
owner_designation to global_id (as id)	product_definition.id	41	130, 135	
owner_designation to unit (as local_units) <SET[0:0]> <no mapping required>				
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.ass	41	129	VERSION_ID<product_definition>

Application Element	AIM Element	Source	Rules	Reference Path
	igned_id			
SHIPTYPE	extended_product_related_product_category	218		<i>see subtype mappings</i>
description	product_category.description	41		extended_product_related_product_category <= product_related_product_category <= product_category product_category.description
shiptype to ship (as defined_for) <SET[1:?]>	PATH			extended_product_related_product_category <= product_related_product_category product_related_product_category.products[i] -> product {CLASS_ID<product, 'ship'>}
- from FUNCTIONAL_DEFINITION				
user_def_function <OPTIONAL>	product_category.name	41		product_category_relationship product_category_relationship.category -> product_category product_category_relationship product_category_relationship.sub_category -> product_category product_category.name
shiptype to unit (as local_units) <SET[0:0]> <no mapping required>				
- from DEFINITION				
shiptype to global_id (as id)	extended_product_related_product_category.id	218	130, 136	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<extended_product_related_product_category>
CARRIER	extended_product_related_product_category	218		{[PROD_CAT_NAME<'carrier'>] [CLASS<extended_product_related_product_category, 'ship type', 'functional definition'>] [CLASS<extended_product_related_product_category,

Application Element	AIM Element	Source	Rules	Reference Path
				'functional definition', 'definition'>] [CLASS<extended_product_related_product_category, 'definition', 'versionable object'>] ROOT_CLASS<extended_product_related_product_category, 'versionable object'>]]}
has_type	product_category.name	41		PROD_CAT_NAME<'carrier' product_category <- product_category_relationship.category product_category_relationship {product_category_relationship.name = 'carrier types'} product_category_relationship.sub_category -> product_category {(product_category.name = 'Container Carrier') (product_category.name = 'Bulk Carrier') (product_category.name = 'Ore Carrier') (product_category.name = 'Oil Tanker') (product_category.name = 'Roro Vessel') (product_category.name = 'Ferry') (product_category.name = 'Car Ferry') (product_category.name = 'Cruise liner') (product_category.name = 'Passenger Vessel') (product_category.name = 'Cargo ship carrying passengers') (product_category.name = 'Product Tanker') (product_category.name = 'Gas carrier') (product_category.name = 'Liquefied Gas Tanker') (product_category.name = 'Chemical Tanker') (product_category.name = 'Chemical Tanker Type 1') (product_category.name = 'Tanker for Refrigerated Fruit Juice') (product_category.name = 'General Cargo Carrier') (product_category.name = 'Dry cargo Vessel') (product_category.name = 'Refrigerated cargo carrying

Application Element	AIM Element	Source	Rules	Reference Path
				ship') (product_category.name = 'High Speed Craft Passenger') (product_category.name = 'High Speed Craft Cargo') (product_category.name = 'Hydrofoil') (product_category.name = 'Car Carrier') (product_category.name = 'Barge') (product_category.name = 'Barge for Deck Loading') (product_category.name = 'Barge for Oil') (product_category.name = 'Barge for Liquefied Gas') (product_category.name = 'Barge Pontoon') (product_category.name = 'user defined')
- from SHIPTYPE				
description				<i>see supertype mappings</i>
carrier to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from FUNCTIONAL_DEFINITION				
user_def_function <OPTIONAL>				<i>see supertype mappings</i>
carrier to unit (as local_units) <SET[0:0]>				<i>see supertype mappings</i>
- from DEFINITION				
carrier to global_id (as id)				<i>see supertype mappings</i>
- from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
WORKING_SHIP	extended_product_related_product_category	218		{[PROD_CAT_NAME<'working ship'>] [CLASS<extended_product_related_product_category, 'ship type', 'functional definition'>] [CLASS<extended_product_related_product_category, 'functional definition', 'definition'>] [CLASS<extended_product_related_product_category, 'definition', 'versionable object'>] ROOT_CLASS<extended_product_related_product_category, 'versionable object'>]}

Application Element	AIM Element	Source	Rules	Reference Path
has_type	product_category.name	41		PROD_CAT_NAME<'working ship'> product_category <- product_category_relationship.category product_category_relationship {product_category_relationship.name = 'working ship types'} product_category_relationship.sub_category -> product_category {(product_category.name = 'Tug') (product_category.name = 'Sealer') (product_category.name = 'Fire Fighter') (product_category.name = 'Drilling Vessel') (product_category.name = 'Pipe Laying Vessel') (product_category.name = 'Crane Vessel') (product_category.name = 'Dredger') (product_category.name = 'Supply Vessel') (product_category.name = 'Ice breaker') (product_category.name = 'Fishing Vessel') (product_category.name = 'Floating Dock') (product_category.name = 'Pilot Boat') (product_category.name = 'Floating Hotel') (product_category.name = 'Well Stimulation Vessel') (product_category.name = 'Pusher') (product_category.name = 'Stern Trawler') (product_category.name = 'Reefer') (product_category.name = 'Offshore Supply Vessel') (product_category.name = 'Oil Production Vessel') (product_category.name = 'Oil Storage Vessel') (product_category.name = 'Oil Production and Storage Vessel') (product_category.name = 'Shuttle tanker') (product_category.name = 'FPSO') (product_category.name = 'FPGO')

Application Element	AIM Element	Source	Rules	Reference Path
				{(product_category.name = 'user defined')}
- from SHIPTYPE				
description				<i>see supertype mappings</i>
working_ship to ship (as defined_for) <SET[1:?]>	PATH			<i>see supertype mappings</i>
- from FUNCTIONAL_DEFINITION				
user_def_function <OPTIONAL>				<i>see supertype mappings</i>
working_ship to unit (as local_units) <SET[0:0]>				<i>see supertype mappings</i>
- from DEFINITION				
working_ship to global_id (as id)				<i>see supertype mappings</i>
- from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
RESEARCH_SHIP	extended_product_related_product_category	218		{[PROD_CAT_NAME<'research ship'>] [CLASS<extended_product_related_product_category, 'ship type', 'functional definition'>] [CLASS<extended_product_related_product_category, 'functional definition', 'definition'>] [CLASS<extended_product_related_product_category, 'definition', 'versionable object'>] ROOT_CLASS<extended_product_related_product_category, 'definition', 'versionable object'>]}
has_type	product_category.name	41		PROD_CAT_NAME<'research ship'> product_category <- product_category_relationship.category product_category_relationship {product_category_relationship.name = 'research ship types'} product_category_relationship.sub_category -> product_category {product_category.name = 'user defined'}
- from SHIPTYPE				

Application Element	AIM Element	Source	Rules	Reference Path
description				<i>see supertype mappings</i>
research_ship to ship (as defined_for) <SET[1:?]>	PATH			<i>see supertype mappings</i>
- from FUNCTIONAL_DEFINITION				
user_def_function <OPTIONAL>				<i>see supertype mappings</i>
research_ship to unit (as local_units) <SET[0:0]>				<i>see supertype mappings</i>
- from DEFINITION				
research_ship to global_id (as id)				<i>see supertype mappings</i>
- from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
NAVY_SHIP	extended_product_related_product_category	218		<pre> {[PROD_CAT_NAME<'navy ship'>] [CLASS<extended_product_related_product_category, 'ship type', 'functional definition'>] [CLASS<extended_product_related_product_category, 'functional definition', 'definition'>] [CLASS<extended_product_related_product_category, 'definition', 'versionable object'>] ROOT_CLASS<extended_product_related_product_category, 'versionable object'>]} </pre>
has_type	product_category.name	41		<pre> PROD_CAT_NAME<'navy ship'> product_category <- product_category_relationship.category product_category_relationship {product_category_relationship.name = 'navy ship types'} product_category_relationship.category -> product_category product_category_relationship product_category_relationship.sub_category -> product_category {(product_category.name = 'AIRCRAFT CARRIER') (product_category.name = 'CORVETTE')} </pre>

Application Element	AIM Element	Source	Rules	Reference Path
				(product_category.name = 'CRUISER') (product_category.name = 'DESTROYER') (product_category.name = 'FLEET AUXILIARY VESSEL') (product_category.name = 'FRIGATE') (product_category.name = 'MINE WARFARE SHIP') (product_category.name = 'PATROL FORCE VESSEL') (product_category.name = 'SERVICE CRAFT') (product_category.name = 'SUBMARINE') (product_category.name = 'AUXILIARY OILER') (product_category.name = 'LANDING PLATFORM DOCK') (product_category.name = 'LANDING PLATFORM HELICOPTER') (product_category.name = 'user defined')
- from SHIPTYPE				
description				<i>see supertype mappings</i>
navy_ship to ship (as defined_for) <SET[1:?]>	PATH			<i>see supertype mappings</i>
- from FUNCTIONAL_DEFINITION				
user_def_function <OPTIONAL>				<i>see supertype mappings</i>
navy_ship to unit (as local_units) <SET[0:0]>				<i>see supertype mappings</i>
- from DEFINITION				
navy_ship to global_id (as id)				<i>see supertype mappings</i>
- from VERSIONABLE_OBJECT				
version_id				<i>see supertype mappings</i>
FREEBOARD_CHARACTERISTICS	product_definition	41	156	{[CLASS<product_definition, 'freeboard characteristics', 'general characteristics definition'>] [CLASS<product_definition, 'general characteristics definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable

Application Element	AIM Element	Source	Rules	Reference Path
				object’>] ROOT_CLASS<product_definition, ‘versionable object’>]}
freeboard_characterics to date_and_time (as date_freeboard_assigned)	PATH		157	DAT_TIME_ASSGN<product_definition, ‘date freeboard assigned’>
freeboard_characterics to organisation (as freeboard_assigned_by)	PATH		158	ORG_ASSGN<product_definition, ‘freeboard assigned by’>
freebord	value_representation_item.va lue_component	43	128, 159	PROD_DEF_TO_VAL_REP_ITEM<‘freeboard characteristics’, ‘freeboard’, positive_length_measure>
freeboard_characterics to loadline (as applicable_loadline)	PATH		160	PROD_DEF_PROP_DEF {CLASS_ID<property_definition, ‘loadline’>}
- from GENERAL_CHARACTERISTICS_DEFINITIO N				
freeboard_characterics to ship (as defined_for) <SET[1:?]>				<i>see supertype mappings</i>
- from DEFINITION				
freeboard_characterics to global_id (as id)	product_definition.id	41	130, 131	
freeboard_characterics to unit (as local_units) <SET>	PATH			PROD_DEF_TO_UNITS<‘freeboard characteristics’>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.ass igned_id	41	129	VERSION_ID<product_definition>
LOADLINE	property_definition	41	161	{CLASS_ID<property_definition, ‘loadline’>}
load_line_length	value_representation_item.va lue_component	43	128, 162	PROP_DEF_TO_VAL_REP_ITEM<‘loadline’, ‘load line length’, positive_length_measure>
load_line_depth	value_representation_item.va lue_component	43	128, 162	PROP_DEF_TO_VAL_REP_ITEM<‘loadline’, ‘load line depth’, positive_length_measure>
load_line_displacement	value_representation_item.va lue_component	43	128, 162	PROP_DEF_TO_VAL_REP_ITEM<‘loadline’, ‘load line displacement’, volume_measure>
load_line_block_coefficient	value_representation_item.va	43	128,	PROP_DEF_TO_VAL_REP_ITEM<‘loadline’, ‘load line

Application Element	AIM Element	Source	Rules	Reference Path
	lue_component		162	block coefficient', ratio_measure>
load_line_regulation	descriptive_representation_item.description	45	162	PROP_DEF_TO_DESC_REP_ITEM<'loadline', 'load line regulation'> {(descriptive_representation_item.description = 'ILLC_1930') (descriptive_representation_item.description = 'ILLC_1966') (descriptive_representation_item.description = 'other')}
LIGHTSHIP_DEFINITION	product_definition	41	163	{[CLASS<product_definition, 'lightship_definition', 'definition'>] [CLASS<product_definition, 'definition', 'versionable object'>] ROOT_CLASS<product_definition, 'versionable object'>]}
lightship_weight	value_representation_item.value_component	43	128, 164	PROD_DEF_TO_VAL_REP_ITEM<'lightship definition', 'lightship weight', mass_measure>
lightship_definition to cartesian_point (as lightship_centre_of_gravity)	PATH	42	128, 164	PROD_DEF_TO_CART_POINT<'lightship centre of gravity'>
lightship_definition to lightship_weight_item (as lightship_items) <SET [0:?]>	PATH			PROD_DEF_PROP_DEF {CLASS_ID<property_definition, 'lightship_weight_item'>}
lightship_definition to ship (as defined_for) <SET [1:1]>	PATH			PROD_DEF_PRODUCT {CLASS_ID<product, 'ship'>}
- from DEFINITION				
lightship_definition to global_id (as id)	product_definition.id	41	130, 131	
lightship_definition to unit (as local_units) <SET>	PATH			PROD_DEF_TO_UNITS<'lightship definition'>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	129	VERSION_ID<product_definition>

Application Element	AIM Element	Source	Rules	Reference Path
LIGHTSHIP_WEIGHT_ITEM	property_definition	41	165	{CLASS_ID<property_definition, 'lightship weight item'>}
lightship_item_description	property_definition.description	41		
aft_weight_extent	value_representation_item.value_component	43	128, 166	PROP_DEF_TO_VAL_REP_ITEM<'lightship weight item', 'aft weight extent', length_measure>
fwd_weight_extent	value_representation_item.value_component	43	128, 166	PROP_DEF_TO_VAL_REP_ITEM<'lightship weight item', 'fwd weight extent', length_measure>
- from WEIGHT_AND_CENTRE_OF_GRAVITY				
mass				<i>see supertype</i>
lightship_weight_item to cartesian_point (as centre_of_gravity)				<i>see supertype</i>
lightship_weight_item to moment_3d (as moment)				<i>see supertype</i>
operating_condition				<i>see supertype</i>
WEIGHT_AND_CENTRE_OF_GRAVITY	property_definition	41	167	{CLASS_ID<property_definition, 'weight and centre of gravity'>}
mass	value_representation_item.value_component	43	128, 168	PROP_DEF_TO_VAL_REP_ITEM<'weight and centre of gravity', 'mass', mass_measure>
weight_and_centre_of_gravity to cartesian_point (as centre_of_gravity)	PATH	42	128, 168	PROP_DEF_TO_CART_POINT<'weight and centre of gravity', 'centre of gravity'>
weight_and_centre_of_gravity to moment_3d (as moment)	PATH		169	PROP_DEF_TO_REP {representation representation.name= 'moment 3d'>}
MOMENT_3D	representation	43		{representation representation.name= 'moment 3d'>}
longitudinal_moment	value_representation_item.value_component	43	128, 170	REP_TO_SPECIAL_VAL_REP_ITEM<'longitudinal moment', moment_unit>
transverse_moment	value_representation_item.value_component	43	128, 170	REP_TO_SPECIAL_VAL_REP_ITEM<'transverse moment', moment_unit>
vertical_moment	value_representation_item.value_component	43	128,	REP_TO_SPECIAL_VAL_REP_ITEM<'vertical moment',

Application Element	AIM Element	Source	Rules	Reference Path
	lue_component		170	moment_unit>
moment_3d to cartesian_point (as origin)	cartesian_point	42	128, 170	REP_TO_CART_POINT<'origin'>

Table 11 - Mapping table for ship_manufacturing_definitions UoF

Application Element	AIM element	Source	Rules	Reference Path
ASSEMBLY_BOUNDING_BOX	representation	43		representation {representation.name='bounding box representation'}
assembly_bounding_box to cartesian_point (as point_min)	PATH		173	representation {representation.name='bounding box representation' representation.items[i] -> GEO_REP_ITEM<'point min', cartesian_point>}}
assembly_bounding_box to cartesian_point (as point_max)	PATH		173	representation {representation.name='bounding_box' representation.items[i] -> GEO_REP_ITEM<'point max', cartesian_point>}}
ASSEMBLY_MANUFACTURING_DEFINITION	extended_product_definition_shape	218	177	{[CLASS<extended_product_definition_shape, 'assembly manufacturing definition', 'manufacturing definition'>] [CLASS<extended_product_definition_shape, 'manufacturing definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
assembly_method	descriptive_representation_item. description	45	176	extended_property_definition <= property_definition = represented_definition <- PDR<'assembly manufacturing definition parameters'> -> representation {[representation.name = '.UNUSED.'] [representation.id = '.UNUSED.']] representation.items[i] - > representation_item {representation_item.name = 'assembly method'} representation_item => descriptive_representation_item descriptive_representation_item.description

Application Element	AIM element	Source	Rules	Reference Path
assembly_stage	descriptive_representation_item. description	45	176	<pre> extended_property_definition <= property_definition = represented_definition <- PDR<'assembly manufacturing definition parameters'> -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'assembly stage'} representation_item => descriptive_representation_item descriptive_representation_item.description </pre>
mass	value_representation_item. value_component	43	173, 176	<pre> extended_property_definition <= property_definition = represented_definition <- PDR<'assembly manufacturing definition parameters'> -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > VAL_REP_ITEM<'mass', mass_measure> </pre>
assembly_manufacturing_definition to date_and_time (as production_date)	date_and_time	41		<pre> {[DAT_TIME_ASSGN <extended_product_definition_shape, 'production date'>]} </pre>
assembly_manufacturing_definition to cartesian_point (as centre_of_gravity)	PATH		173, 176, 186	<pre> extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'centre_of_gravity'>]} [shape_aspect = represented_definition <- PDR<'centre of gravity representation'> -> representation </pre>

Application Element	AIM element	Source	Rules	Reference Path
assembly_manufacturing_definition to assembly (as defined_for) (SET[1:?])	PATH			representation.items[i] -> GEO_REP_ITEM<'.UNUSED.', cartesian_point>]}
assembly_manufacturing_definition to assembly (as defined_for) (SET[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS<product_definition, 'assembly'>}
assembly_manufacturing_definition to assembly_manufacturing_position (as assembly_positions) (SET[0:?])	PATH			extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {CLASS<shape_aspect, 'assembly manufacturing position'>
assembly_manufacturing_definition to document_reference (as assembly_drawing)	PATH			extended_product_definition_shape = document_reference_item <= applied_document_reference.items[i] applied_document_reference {applied_document_reference <= document_reference document_reference.role -> object_role [object_role.name = 'assembly drawing'] [object_role.description = '.UNUSED. ']}
- from MANUFACTURING_DEFINITION				
assembly_manufacturing_definition to shape_representation (as representations) (SET[0:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition <= represented_definition <= PDR<'assembly manufacturing definition shape representation'> property_definition_representation.used_representation ->

Application Element	AIM element	Source	Rules	Reference Path
				representation => shape_representation
- from DEFINITION				
assembly_manufacturing_definition to global_id (as id)	extended_product_definition _shape.id	218	172, 174	
assembly_manufacturing_definition to unit (as local_units) (SET [0:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition {property_definition.name = 'UNUSED.'} property_definition <= represented_definition <= property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = 'UNUSED.']}
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id		171	extended_product_definition_shape <= applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'}

Application Element	AIM element	Source	Rules	Reference Path
				identification_assignment.assigned_id
ASSEMBLY_MANUFACTURING_POSITION	shape_aspect	41	179	{[ROOT_CLASS<shape_aspect, 'assembly manufacturing position'>]}
assembly_manufacturing_position to bounded_curve (as assembly_footprint)	bounded_curve	42	173, 178	<pre> shape_aspect = represented_definition <- PDR<'assembly manufacturing position parameters'> ->representation {representation.name = 'assembly footprint representation'} representation.items [i] -> representation_item {representation_item.name = 'assembly_footprint'} => geometric_representation_item => bounded_curve </pre>
assembly_manufacturing_position to assembly_bounding_box (as bounding_box)	PATH			<pre> shape_aspect = represented_definition <- PDR<'assembly manufacturing position parameters'> ->representation => shape_representation {representation.name = 'bounding box representation'} </pre>
assembly_manufacturing_position to direction (as orientation)	direction	42	173, 178	<pre> shape_aspect = represented_definition <- PDR<'assembly manufacturing position parameters'> ->representation {representation.name = 'orientation representation'} representation.items [i] -> representation_item {representation_item.name = 'orientation'} => geometric_representation_item => direction </pre>
PLATE_MANUFACTURING_DEFINITION	extended_product_definition _shape	218	180	{[CLASS<extended_product_definition_shape, 'plate manufacturing definition', 'structural part manufacturing definition'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<extended_product_definition_shape, 'structural part manufacturing definition', 'manufacturing definition'>] [CLASS<extended_product_definition_shape, 'manufacturing definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]
plate_manufacturing_definition to bounded_curve (as outer_contour) (SET[1:?])	PATH		173, 181	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'outer contour'>] [shape_aspect = represented_definition <= PDR<'outer contour representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
plate_manufacturing_definition to structural_part (as defined_for) #1: structural_part is a Plate #2: structural_part is a Corrugated_part (SET[1:?])	#1: PATH #2: PATH			#1: extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition { CLASS_ID<product_definition, 'plate'>} #2: extended_product_definition_shape <= product_definition_shape <= property_definition

Application Element	AIM element	Source	Rules	Reference Path
				property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'corrugated_part'>}
- from STRUCTURAL_PART_MANUFACTURING_ DEFINITION				
plate_manufacturing_definition to curve (as annotation)	<i>See supertype</i>			
plate_manufacturing_definition to curve (as inner_contours)	<i>See supertype</i>			
plate_manufacturing_definition to curve (as layout_marks)	<i>See supertype</i>			
- from MANUFACTURING_DEFINITION				
plate_manufacturing_definition to shape_representation (as representations)	<i>see supertype</i>			
- from DEFINITION				
plate_manufacturing_definition to global_id (as id)	<i>see supertype</i>			
plate_manufacturing_definition to unit (as local_units)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROFILE_MANUFACTURING_DEFINITION	extended_product_definition _shape	218		{[CLASS<extended_product_definition_shape, 'profile manufacturing definition', 'structural part manufacturing definition'>] [CLASS<extended_product_definition_shape, 'structural part manufacturing definition', 'manufacturing definition'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<extended_product_definition_shape, 'manufacturing definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]
profile_manufacturing_definition to bounded_curve (as inverse_bend_trace) (SET[0:?])	PATH		173	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'inverse bend trace'>] [shape_aspect = represented_definition <= PDR<'inverse bend trace representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
profile_manufacturing_definition to bounded_curve (as outer_flange_countour) (SET[0:?])	PATH		173	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'outer flange contour'>] [shape_aspect = represented_definition <= PDR<'outer flange contour representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
profile_manufacturing_definition to	PATH		173	extended_product_definition_shape <=

Application Element	AIM element	Source	Rules	Reference Path
bounded_curve (as outer_web_countour) (SET[0:?])				product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'outer web contour'] [shape_aspect = represented_definition <- PDR<'outer web contour representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
profile_manufacturing_definition to profile (as defined_for) (SET[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'profile'>}
- from STRUCTURAL_PART_MANUFACTURING_ DEFINITION				
structural_part_manufacturing_definition to curve (as annotation)	<i>See supertype</i>			
structural_part_manufacturing_definition to curve (as inner_contours)	<i>See supertype</i>			
structural_part_manufacturing_definition to curve (as layout_marks)	<i>See supertype</i>			
- from MANUFACTURING_DEFINITION				
profile_manufacturing_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
shape_representation (as representations) (SE)				
- from DEFINITION				
profile_manufacturing_definition to global_id (as id)	<i>see supertype</i>			
profile_manufacturing_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
STRUCTURAL_ADDED_MATERIAL_BOUNDARY_RELATIONSHIP	extended_shape_aspect_relationship	218		extended_shape_aspect_relationship {[CLASS<extended_shape_aspect_relationship, 'structural added material boundary relationship', 'structural feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'structural feature relationship', 'feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'feature relationship', 'item relationship'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'definable object'>] [CLASS<extended_shape_aspect_relationship, 'definable object', 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'versionable object'>]}<= shape_aspect_relationship
boundary_index	shape_aspect	41		extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] [{shape_aspect.of_shape -> product_definition_shape <= <PROP_TO_PROD_DEF> {(CLASS_ID<product_definition, 'profile'>)}

Application Element	AIM element	Source	Rules	Reference Path
				(CLASS_ID<product_definition, 'plate'>) (CLASS_ID<product_definition, 'corrugated part'>)]
structural_added_material_boundary_relationship to structural_added_material_feature (as item_1) (OPT)	PATH			extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.relate_shape_aspect -> shape_aspect => extended_shape_aspect (CLASS_ID<extended_shape_aspect, 'structural added material feature'>)
- from STRUCTURAL_FEATURE_RELATIONSHIP				
- from FEATURE_RELATIONSHIP				
- from ITEM_RELATIONSHIP				
structural_added_material_boundary_relationship to external_instance_reference (as external_item_1) (OPT)	PATH			extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'extended shape aspect'>
structural_added_material_boundary_relationship to external_instance_reference (as external_item_2) (OPT)	PATH			extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'product definition'> {(CLASS_ID<product_definition, 'profile'>) (CLASS_ID<product_definition, 'plate'>) (CLASS_ID<product_definition, 'corrugated part'>))}
structural_added_material_boundary_relationship to item (as item_2) #1: item is a Profile #2: item is a Plate #3: item is a Corrugated_part (OPT)	#1: PATH #2: PATH #3: PATH			#1: extended_shape_aspect_relationship <= shape_aspect_relationship <= shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] shape_aspect.of_shape -> product_definition_shape <= PROP_TO_PROD_DEF {(CLASS_ID<product_definition, 'profile'>))}

Application Element	AIM element	Source	Rules	Reference Path
				#2: extended_shape_aspect_relationship <= shape_aspect_relationship <- shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] shape_aspect.of_shape -> product_definition_shape <= PROP_TO_PROD_DEF { (CLASS_ID<product_definition, 'plate'>) } #3: extended_shape_aspect_relationship <= shape_aspect_relationship <- shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] shape_aspect.of_shape -> product_definition_shape <= PROP_TO_PROD_DEF { (CLASS_ID<product_definition, 'corrugated part'>) }
contxt (OPT)	extended_shape_aspect_relationship.description	218		
- from DEFINABLE_OBJECT				
structural_added_material_boundary_relationship to global_id (as id)	extended_shape_aspect_relationship.id	218	172 175	
structural_added_material_boundary_relationship to definition (as definitions) (INV) Issue: Will there ever be definitions assigned to this thing?				
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.assigned_id	41	171	extended_shape_aspect_relationship HAS_ID<extended_shape_aspect_relationship, 'version

Application Element	AIM element	Source	Rules	Reference Path
				id> identification_assignment.assigned_id
STRUCTURAL_ADDED_MATERIAL_FEATURE	extended_shape_aspect	218		extended_shape_aspect {[CLASS<extended_shape_aspect, 'structural added material feature', 'structural manufacturing feature'>] [CLASS<extended_shape_aspect, 'structural manufacturing feature', 'structural feature'>] [CLASS<extended_shape_aspect, 'structural feature', 'feature'>] [CLASS<extended_shape_aspect, 'feature', 'item'>] [CLASS<extended_shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<extended_shape_aspect, 'definable object'>]]<= shape_aspect
structural_added_material_feature to item (as parent) #1: item is a Plate #2: item is a Profile #3: item is a Corrugated_part	#1: PATH #2: PATH #3: PATH			#1: extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {(CLASS_ID<product_definition, 'plate'>)} #2: extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'}

Application Element	AIM element	Source	Rules	Reference Path
				property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {(CLASS_ID<product_definition, 'profile'>)} #3: extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {(CLASS_ID<product_definition, 'corrugated part'>)}
- from STRUCTURAL_MANUFACTURING_FEATU RE				
- from STRUCTURAL_FEATURE				
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
structural_added_material_feature to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
structural_added_material_feature to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
structural_added_material_feature to global_id (as id)	<i>see supertype</i>			
structural_added_material_feature to structural_added_material_feature_design_definition (as definitions) (INV)	PATH			extended_shape_aspect <= shape_aspect SA2REP_DN<'structural added material feature design definition parameter representation', 'defined_for'> representation {CLASS_ID<representation, 'structural added material feature design definition'>}
STRUCTURAL_ADDED_MATERIAL_FEATURE_DESIGN_DEFINITION	representation	43		representation {[CLASS<representation, 'structural added material feature design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}
added_material_length	value_representation_item.value_component	43	173, 182	representation representation.items [i] -> VAL_REP_ITEM<'added material length', positive_length_measure>
structural_added_material_feature_design_definition to structural_added_material_feature (as defined_for) (SET[1:?])	PATH			representation REP2SA_DN<'structural added material feature design definition parameter representation', 'defined_for'> shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'structural added material feature'>)}
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
structural_added_material_feature_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from DEFINITION				
structural_added_material_feature_design_definition to global_id (as id)	<i>see supertype</i>			
structural_added_material_feature_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
STRUCTURAL_MANUFACTURING_FEATURE #1: structural_added_material_feature #2: structural_weld_shrinkage_allowance_feature	extended_shape_aspect	218		<i>see subtypes</i>
structural_manufacturing_feature to item (as parent)	PATH			extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'structural part'>}
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
structural_manufacturing_feature to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
description (OPT)	<i>see supertype</i>			
structural_manufacturing_feature to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
structural_manufacturing_feature to global_id (as id)	<i>see supertype</i>			
structural_manufacturing_feature to definition (as definitions) (INVERSE) (SET[0:?])	<i>see subtypes</i>			
STRUCTURAL_PART_MANUFACTURING_ DEFINITION #1: plate_manufacturing_definition #2: profile_manufacturing_definition	extended_product_definition _shape	218		<i>see subtypes</i>
structural_part_manufacturing_definition to bounded_curve (as annotation) SET[0:?]	PATH		173	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'annotation'>] [shape_aspect = represented_definition <= PDR<'annotation representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
structural_part_manufacturing_definition to	PATH		173	extended_product_definition_shape <=

Application Element	AIM element	Source	Rules	Reference Path
bounded_curve (as inner_contours) SET[0:?]				product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'inner countour'] [shape_aspect = represented_definition <- PDR<'inner countour representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
structural_part_manufacturing_definition to bounded_curve (as layout_marks) SET[0:?]	PATH		173	extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'layout mark'] [shape_aspect = represented_definition <- PDR<'layout mark representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]}
- from MANUFACTURING_DEFINITION				
structural_part_manufacturing_definition to shape_representation (as representations) SET[0:?]	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition <- represented_definition <- PDR<'representation' property_definition_representation.used_representation -> representation =>

Application Element	AIM element	Source	Rules	Reference Path
				shape_representation
structural_part_manufacturing_definition to structural_part (as defined_for)				<i>see subtypes</i>
- from DEFINITION				
structural_part_manufacturing_definition to global_id (as id)	extended_product_definition_shape.id	218	172 174	
structural_part_manufacturing_definition to unit (as local_units)	PATH			<pre> extended_product_definition_shape <= product_definition_shape <= property_definition {property_definition.name = 'UNUSED.'} property_definition <- represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = 'UNUSED.']} </pre>
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment.assigned_id		171	<pre> extended_product_definition_shape <- applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> identification_role.name = 'version id' identification_assignment.assigned_id </pre>
STRUCTURAL_WELD_SHRINKAGE_ALLOWANCE_FEATURE	extended_shape_aspect	218		<pre> extended_shape_aspect {[CLASS<extended_shape_aspect, 'structural weld shrinkage allowance feature', 'structural manufacturing feature'>]} [CLASS<extended_shape_aspect, 'structural manufacturing feature', 'structural feature'>] [CLASS<extended_shape_aspect, 'structural feature', 'feature'>] [CLASS<extended_shape_aspect, 'feature', 'item'>] [CLASS<extended_shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<extended_shape_aspect, 'definable object'>]]<= shape_aspect </pre>
<p>structural_weld_shrinkage_allowance_feature to item (as parent)</p> <p>#1: item is Plate</p> <p>#2: item is Profile</p>	<p>#1: PATH</p> <p>#2: PATH</p>			<pre> #1: extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {((CLASS_ID<product_definition, 'plate'>))} #2: extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition </pre>

Application Element	AIM element	Source	Rules	Reference Path
				{property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {(CLASS_ID<product_definition, 'profile'>)}
- from STRUCTURAL_MANUFACTURING_FEATU RE				
- from STRUCTURAL_FEATURE				
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
structural_weld_shrinkage_allowance_feature to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
structural_weld_shrinkage_allowance_feature to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
structural_weld_shrinkage_allowance_feature to global_id (as id)	<i>see supertype</i>			
structural_weld_shrinkage_allowance_feature to structural_weld_shrinkage_allowance_feature_d esign_definition (as definitions) (INV)	PATH			extended_shape_aspect <= shape_aspect SA2REP_DN<'structural weld shrinkage allowance feature design definition parameter representation', 'defined_for'> representation {CLASS_ID<representation, 'structural weld shrinkage allowance feature design definition'>}
STRUCTURAL_WELD_SHRINKAGE_ALLO WANCE_-FEATURE_DESIGN_DEFINITION	representation	43		representation {[CLASS<representation, 'structural weld shrinkage

Application Element	AIM element	Source	Rules	Reference Path
				allowance feature design definition', 'feature design definition'] [CLASS<representation, 'feature design definition', 'design definition'] [CLASS<representation, 'design definition', 'definition'] [CLASS<representation, 'definition', 'versionable object'] [ROOT_CLASS<representation, 'versionable object']}]
longitudinal_allowance (OPT)	value_representation_item.v alue_component	43	173, 183	representation representation.items [i] -> VAL_REP_ITEM<'longitudinal allowance', ratio_measure>
transverse_allowance (OPT)	value_representation_item.v alue_component	43	173, 183	representation representation.items [i] -> VAL_REP_ITEM<'transverse allowance', ratio_measure>
vertical_allowance (OPT)	value_representation_item.v alue_component	43	173, 183	representation representation.items [i] -> VAL_REP_ITEM<'vertical allowance', ratio_measure>
structural_weld_shrinkage_allowance_feature_d esign_definition to structural_weld_shrinkage_allowance_feature (as defined_for) (SET[1:?])	PATH			representation REP2SA_DN<'structural weld shrinkage allowance feature design definition parameter representation', 'defined_for'> shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'structural weld shrinkage allowance feature'>)}
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
structural_weld_shrinkage_allowance_feature_d esign_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
structural_weld_shrinkage_allowance_feature_d esign_definition to global_id (as id)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
structural_weld_shrinkage_allowance_feature_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
WELD_MANUFACTURING_DEFINITION	extended_product_definition_shape	218		{[CLASS<extended_product_definition_shape, 'weld manufacturing definition', 'manufacturing definition'>] [CLASS<extended_product_definition_shape, 'manufacturing definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
weld_manufacturing_definition to vector (as torch_vector)	PATH		173, 184	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'torch vector'>] [shape_aspect = represented_definition <= PDR<'torch vector representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', vector>]]}
position	descriptive_representation_item. description	45	185	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'weld manufacturing definition parameters'> -> representation {[representation.name = 'UNUSED.' [representation.id = 'UNUSED.']] representation.items[i] -

Application Element	AIM element	Source	Rules	Reference Path
				> representation_item {representation_item.name = 'position'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'flat') (descriptive_representation_item.description = 'horizontal') (descriptive_representation_item.description = 'vertical') (descriptive_representation_item.description = 'overhead') (descriptive_representation_item.description = 'other')}
process	descriptive_representation_item. description	45	185	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'weld manufacturing definition parameters'> -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'process'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'SMAW') (descriptive_representation_item.description = 'SAW') (descriptive_representation_item.description = 'GMAW') (descriptive_representation_item.description = 'FCAW') (descriptive_representation_item.description = 'user defined')}
degree_of_automations	descriptive_representation_item. description	45	185	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'weld manufacturing definition parameters'> -> representation

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'degree of automations'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'manual') (descriptive_representation_item.description = 'self run') (descriptive_representation_item.description = 'automatic') (descriptive_representation_item.description = 'robotic') (descriptive_representation_item.description = 'mechanized') (descriptive_representation_item.description = 'user defined')} </pre>
number_of_weld_passes	value_representation_item. value_component	43	173, 185	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'weld manufacturing definition parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'number of weld passes', count_measure> </pre>
welding_deposition_sequences	descriptive_representation_i tem. description	45	185	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'weld manufacturing definition parameters'> -> representation {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] - > representation_item </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {representation_item.name = 'welding deposition sequences'} representation_item => descriptive_representation_item { (descriptive_representation_item.description = 'progressive') (descriptive_representation_item.description = 'backstep') (descriptive_representation_item.description = 'wandering') (descriptive_representation_item.description = 'user defined')} </pre>
weld_manufacturing_definition to weld_testing (as weld_test) SET[1:?]	PATH			<pre> extended_product_definition_shape = action_assignment_item <- applied_action_assignment.items[i] applied_action_assignment <= action_assignment { action_assignment.role -> object_role [object_role.name = 'weld test'] [object_role.description = '.UNUSED. ']} action_assignment.assigned_action -> executed_action <= action </pre>
weld_manufacturing_definition to welding_sequence (as sequences) SET[1:?]	PATH	45		<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'welding sequence representation'> -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} </pre>
weld_manufacturing_definition to document (as welding_procedures) SET[1:?]	PATH			<pre> extended_product_definition_shape = document_reference_item <- applied_document_reference.items[i] applied_document_reference </pre>

Application Element	AIM element	Source	Rules	Reference Path
				{applied_document_reference <= document_reference document_reference.role -> object_role [object_role.name = 'welding procedures'] [object_role.description = 'UNUSED. ']}
welding_environment	descriptive_representation_item. description	45	185	extended_property_definition <= property_definition = represented_definition <- PDR<'weld manufacturing definition parameters'> -> representation {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'welding environment'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'field') (descriptive_representation_item.description = 'shop')}
tack_weld_used	descriptive_representation_item. description	45	185	extended_property_definition <= property_definition = represented_definition <- PDR<'weld manufacturing definition parameters'> -> representation {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'tack weld used'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
weld_manufacturing_definition to weld	PATH			extended_product_definition_shape <=

Application Element	AIM element	Source	Rules	Reference Path
(as defined_for) SET[1:?]				product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'weld'>}
- from MANUFACTURING_DEFINITION				
weld_manufacturing_definition to shape_representation (as representations) SET[0:?]	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition <= represented_definition <= PDR<'representation'> property_definition_representation.used_representation -> representation => shape_representation
- from DEFINITION				
weld_manufacturing_definition to global_id (as id)	extended_product_definition _shape.id	218	172 174	
weld_manufacturing_definition to unit (as local_units) SET[0:?]	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition {property_definition.name = 'UNUSED.'} property_definition <= represented_definition <= property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.context_of_items ->

Application Element	AIM element	Source	Rules	Reference Path
				representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']}
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id		171	extended_product_definition_shape <- applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
WELD_TESTING	executed_action	41		
test_methods BAG [1:?]	PATH			executed_action <= action action.chosen_method -> action_method action_method {(action_method.name = 'visual examination') (action_method.name = 'dye penetrant') (action_method.name = 'magnetic particle') (action_method.name = 'radiographic') (descriptive_representation_item.description = 'ultrasonic')} action_method <- action_method_relationship.related_method action_method_relationship action_method_relationship.relate_method -> action_method {(action_method.name = 'visual examination') (action_method.name = 'dye penetrant') (action_method.name = 'magnetic particle')}

Application Element	AIM element	Source	Rules	Reference Path
				(action_method.name = 'radiographic') (descriptive_representation_item.description = 'ultrasonic')}
weld_testing to document (as test_sequence) S[0:?]	PATH			executed_action = document_reference_item <- applied_document_reference.items[i] applied_document_reference {applied_document_reference <= document_reference document_reference.role -> object_role [object_role.name = 'test sequence'] [object_role.description = '.UNUSED. ']}
test_results	PATH			executed_action <- action_status.assigned_action action_status {(action_status.status = 'pre qualified') (action_status.status = 'rejected') (action_status.status = 'pending') (action_status.status = 'other')}
WELDING_SEQUENCE	representation	41		representation {[representation.name = '.UNUSED.'] [representation.id = '.UNUSED.']} <- PDR<'welding sequence representation'> -> represented_definition = extended_product_definition_shape {CLASS_ID<extended_product_definition_shape, 'weld_manufacturing_definition'>}
name	descriptive_representation_item. description	45	187	representation {[representation.name = '.UNUSED.'] [representation.id = '.UNUSED.']} representation.items[i] - > representation_item

Application Element	AIM element	Source	Rules	Reference Path
				{representation_item.name = 'name'} representation_item => descriptive_representation_item
start_offset	value_representation_item. value_component	43	173, 187	representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > VAL_REP_ITEM<'start offset', length_measure>
end_offset	value_representation_item. value_component	43	173, 187	representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > VAL_REP_ITEM<'end offset', length_measure>
direction	descriptive_representation_i tem. description	45	173, 187	representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'tack weld used'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}

Table 12 - Mapping table for ship_materials UoF

Application Element	AIM element	Source	Rules	Reference Path
HOMOGENEOUS_SHIP_MATERIAL_PROPERTY	(extended_property_definition) (extended_material_designation)	(218) (218)	195	(([CLASS<extended_property_definition, 'homogeneous ship material property', 'ship material property '>] [CLASS<extended_property_definition, 'ship material property', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition, 'versionable object'>]]) ({CLASS<extended_material_designation, 'homogeneous ship material'>}))
poisson_ratio	value_representation_item. value_component	43	190, 196	(extended_property_definition <= property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'poisson ratio', ratio_measure>) (extended_material_designation <= material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation {property_definition_representation = 'material parameters'} property_definition_representation.used_representation -> representation

Application Element	AIM element	Source	Rules	Reference Path
				representation.items[i] -> VAL_REP_ITEM<'poisson ratio', ratio_measure>)
stress_of_fracture	value_representation_item. value_component	43	190, 196	(extended_property_definition <= property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'stress of fracture', context_dependent_measure>) (extended_material_designation <= material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation {property_definition_representation = 'material parameters'} property_definition_representation.used_representation -> representation representation.items[i] -> VAL_REP_ITEM<'stress of fracture', context_dependent_measure>)
thermal_expansion_coefficient	value_representation_item. value_component	43	190, 196	(extended_property_definition <= property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'thermal expansion coefficient', context_dependent_measure>)

Application Element	AIM element	Source	Rules	Reference Path
				(extended_material_designation <= material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation {property_definition_representation = 'material parameters'} property_definition_representation.used_representation -> representation representation.items[i] -> VAL_REP_ITEM<'thermal expansion coefficient', context_dependent_measure>))
yield_point	value_representation_item. value_component	43	190, 196	(extended_property_definition <= property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'yield point', context_dependent_measure>) (extended_material_designation <= material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation {property_definition_representation = 'material parameters'}

Application Element	AIM element	Source	Rules	Reference Path
				property_definition_representation.used_representation -> representation representation.items[i] -> VAL_REP_ITEM<'yield point', context_dependent_measure>)
youngs_module	value_representation_item. value_component	43	190, 196	(extended_property_definition <= property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'youngs module', context_dependent_measure>) (extended_material_designation <= material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation {property_definition_representation = 'material parameters'} property_definition_representation.used_representation -> representation representation.items[i] -> VAL_REP_ITEM<'youngs module', context_dependent_measure>)
homogeneous_ship_material_property to part (as defined_for)	PATH			(extended_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition =

Application Element	AIM element	Source	Rules	Reference Path
				product_definition {CLASS<product_definition, 'part'>}} (extended_material_designation <= material_designation material_designation.definitions[i] -> characterized_definition = characterized_product_definition = product_definition {CLASS<product_definition, 'part'>}}
- from SHIP_MATERIAL_PROPERTY				
density	value_representation_item. value_component	43	190, 196	(extended_property_definition <= property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'density', context_dependent_measure>) (extended_material_designation <= material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation {property_definition_representation = 'material parameters'}) property_definition_representation.used_representation -> representation representation.items[i] -> VAL_REP_ITEM<'density', context_dependent_measure>)

Application Element	AIM element	Source	Rules	Reference Path
description	<i>see supertype</i>			
homogeneous_ship_material_property to document_reference_with_address (as material_reference)	PATH		197	(extended_property_definition = documentation_reference_item <- applied_documentation_reference.items[i] applied_documentation_reference { applied_documentation_reference <= documentation_reference documentation_reference.role -> object_role object_role.name = 'material reference'}) (extended_material_designation = documentation_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role object_role.name = 'material reference'})
- from DEFINITION				
ship_material_property to global_id (as id)	<i>see supertype</i>			
ship_material_property to unit (as local_units)	PATH			(extended_property_definition <= property_definition { property_definition.name = 'UNUSED.'} property_definition = represented_definition <- PDR<'homogeneous ship material property parameters'> -> representation representation.context_of_items -> representation_context => global_unit_assigned_context { global_unit_assigned_context <= representation_context

Application Element	AIM element	Source	Rules	Reference Path
				<pre> [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']} global_unit_assigned_context.units -> unit) (extended_material_designation <= material_designation {material_designation.name = '.UNUSED.'} material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation property_definition_representation.used_representation -> representation representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']})) </pre>
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
SHIP_MATERIAL_PROPERTY	(extended_property_definition) (extended_material_designation)	(218) (218)	192	<pre> ([CLASS<extended_property_definition, 'ship material property', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition, 'versionable object'>]]) ({CLASS<extended_material_designation, </pre>

Application Element	AIM element	Source	Rules	Reference Path
density	value_representation_item. value_component	43	190, 193	'ship material'> }) (extended_property_definition <= property_definition = represented_definition <= PDR<'ship material property parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'density', context_dependent_measure>) (extended_material_designation <= material_designation <= material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation { property_definition_representation = 'material parameters' } property_definition_representation.used_representation -> representation representation.items[i] -> VAL_REP_ITEM<'density', context_dependent_measure>)
description	(property_definition.description ion) (extended_material_designation. ion. Description)	(41) (218)		(extended_property_definition <= property_definition property_definition.description)
ship_material_property to document_reference_with_address (as material_reference)	PATH		194	(extended_property_definition = documentation_reference_item <= applied_document_reference.items[i] applied_document_reference

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {applied_document_reference <= document_reference document_reference.role -> object_role object_role.name = 'material reference'}} (extended_material_designation = documentation_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> object_role object_role.name = 'material reference'}} </pre>
- from DEFINITION				
ship_material_property to global_id (as id)	(extended_property_definition.id) (extended_material_designation.id)	(218) (218)	188, 191	
ship_material_property to unit (as local_units)	PATH			<pre> (extended_property_definition <= property_definition {property_definition.name = '.UNUSED.'} property_definition = represented_definition <- PDR<'ship material property parameters'> -> representation representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']} </pre>

Application Element	AIM element	Source	Rules	Reference Path
				global_unit_assigned_context.units -> unit) (extended_material_designation <= material_designation { material_designation.name = '.UNUSED.'} material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation <= property_definition_representation property_definition_representation.used_representation -> representation representation.context_of_items -> representation_context => global_unit_assigned_context { global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']}
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	189	(HAS_ID<extended_property_definition, 'version_id'> identification_assignment.assigned_id) (extended_material_designation <- applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id)

Table 13 - Mapping table for ship_measures UoF

Application Element	AIM element	Source	Rules	Reference Path
ANGULAR_STIFFNESS_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'angular stiffness measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
angular_stiffness_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'angular stiffness unit'}
ANGULAR_STIFFNESS_UNIT	derived_unit	41		DERIVED_UNIT_4<'angular stiffness unit', mass_unit, 1.0, length_unit, 1.0, time_unit, -2.0, plane_angle_unit, -1.0>
- from DERIVED_UNIT				
angular_stiffness_unit to derived_unit_element (as elements) (SET [1:?])	PATH		198	derived_unit derived_unit.elements[i] -> derived_unit_element
AREA_MEASURE_WITH_UNIT	area_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		area_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = area_measure}
area_measure_with_unit to named_unit (as unit_component)	PATH			area_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { =>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> ([si_unit] [area_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => area_measure_with_unit)}</pre>
AREA_UNIT	area_unit	41		
- from NAMED_UNIT				
area_unit to dimensional_exponents (as dimensions)	PATH			<pre> area_unit <= named_unit named_unit.dimensions -> dimensional_exponents</pre>
DENSITY_MEASURE_WITH_UNIT	measure_with_unit	41		<pre> {[CLASS<measure_with_unit, 'density measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]}</pre>
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		<pre> {measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}</pre>
density_measure_with_unit to derived_unit (as unit_component)	PATH			<pre> measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'density unit'}</pre>
DENSITY_UNIT	derived_unit	41		DERIVED_UNIT_2<'density unit', length_unit, -3.0, mass_unit, 1.0>
- from DERIVED_UNIT				
density_unit to derived_unit_element (as elements) (SET [1:?])	PATH		199	<pre> derived_unit derived_unit.elements[i] -> derived_unit_element</pre>
DERIVED_UNIT	derived_unit	41		
derived_unit to derived_unit_element (as elements)	PATH			<pre> derived_unit derived_unit.elements[i] -></pre>

Application Element	AIM element	Source	Rules	Reference Path derived_unit_element
DERIVED_UNIT_ELEMENT	derived_unit_element	41		
exponent	derived_unit_element.exponent	41		
derived_unit_element to named_unit (as unit)	PATH			derived_unit_element derived_unit_element.unit -> named_unit
DILATATION_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'dilatation measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
dilatation_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'dilatation unit'}
DILATATION_UNIT	derived_unit	41		DERIVED_UNIT_2<'dilatation unit', length_unit, 1.0, thermodynamic_temperature_unit, -1.0>
- from DERIVED_UNIT				
dilatation_unit to derived_unit_element (as elements) (SET [1:?])	PATH		200	derived_unit derived_unit.elements[i] -> derived_unit_element
DIMENSIONAL_EXPONENTS	dimensional_exponents	41		
length_exponent	dimensional_exponents.length_exponent			
mass_exponent	dimensional_exponents.mass_exponent			
time_exponent	dimensional_exponents.time_exponent			
electric_current_exponent	dimensional_exponents.electric_current_exponent			

Application Element	AIM element	Source	Rules	Reference Path
thermodynamic_temperature_exponent	dimensional_exponents.thermodynamic_temperature_exponent			
amount_of_substance_exponent	dimensional_exponents.amount_of_substance_exponent			
luminous_intensity_exponent	dimensional_exponents.luminous_intensity_exponent			
EXTENDED_CONVERSION_BASED_UNIT	conversion_based_unit	41		
name	conversion_based_unit.name	41		
extended_conversion_based_unit to extended_measure_with_unit (as conversion_factor)	PATH			conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit
- from NAMED_UNIT extended_conversion_based_unit to dimensional_exponents (as dimensions)	<i>see supertype</i>			
EXTENDED_GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	global_uncertainty_assigned_context	43		{[CLASS<global_uncertainty_assigned_context, 'extended global uncertainty assigned context', 'representation context'>] [ROOT_CLASS<representation_context, 'representation context'>]}
extended_global_uncertainty_assigned_context to extended_uncertainty_measure_with_unit (as uncertainty) (SET [1:?])	PATH			global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i] -> uncertainty_measure_with_unit {CLASS_ID<uncertainty_measure_with_unit, 'extended uncertainty measure with unit'>}
- from REPRESENTATION_CONTEXT				
context_identifier	representation_context.identifier	43		
context_type	representation_context.type	43		
EXTENDED_MEASURE_WITH_UNIT	measure_with_unit	41		
value_component	measure_with_unit.value_component	41		<i>see subtypes</i>
extended_measure_with_unit to unit (as unit_component)	PATH			<i>see subtypes</i>
EXTENDED_UNCERTAINTY_MEASURE_WITH	uncertainty_measure_with_unit	43		{[CLASS<uncertainty_measure_with_unit, 'extended uncertainty measure with unit', 'extended measure with unit'>]}

Application Element	AIM element	Source	Rules	Reference Path
_UNIT				[ROOT_CLASS<uncertainty_measure_with_unit, 'extended measure with unit'>]]
name	uncertainty_measure_with_unit. name			
description	uncertainty_measure_with_unit. description		213	
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent			{measure_with_unit.value_component -> measure_value measure_value = numeric_measure}
extended_uncertainty_measure_with_unit to unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit {(unit = derived_unit derived_unit) (unit = named_unit named_unit)}
FLOW_RATE_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'flow rate measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
flow_rate_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'flow rate unit'}
FLOW_RATE_UNIT	derived_unit	41		DERIVED_UNIT_2<'flow rate unit', length_unit, 3.0, time_unit, -1.0>
- from DERIVED_UNIT				
flow_rate_unit to derived_unit_element (as elements) (SET [1:?])	PATH		201	derived_unit derived_unit.elements[i] ->

Application Element	AIM element	Source	Rules	Reference Path
				derived_unit_element
FORCE_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'force measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
force_measure_with_unit to unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit {(unit = named_unit named_unit => conversion_based_unit) (unit = derived_unit derived_unit.name = 'force unit')}
FORCE_UNIT	derived_unit	41		DERIVED_UNIT_3<'force unit', mass_unit, 1.0, length_unit, 1.0, time_unit, -2.0>
- from DERIVED_UNIT				
force_unit to derived_unit_element (as elements) (SET [1:?])	PATH		202	derived_unit derived_unit.elements[i] -> derived_unit_element
FREQUENCY_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'frequency measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
frequency_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'frequency unit'}

Application Element	AIM element	Source	Rules	Reference Path
FREQUENCY_UNIT	derived_unit	41		DERIVED_UNIT_1<'frequency unit', time_unit, -1.0>
- from DERIVED_UNIT				
frequency_unit to derived_unit_element (as elements) (SET [1:?])	PATH		203	derived_unit derived_unit.elements[i] -> derived_unit_element
HEAT_RATE_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'heat rate measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
heat_rate_measure_with_unit to unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit {(unit = named_unit named_unit => conversion_based_unit) (unit = derived_unit derived_unit.name = 'heat rate unit')}
HEAT_RATE_UNIT	derived_unit	41		DERIVED_UNIT_3<'heat rate unit', mass_unit, 1.0, length_unit, 2.0, time_unit, -3.0>
- from DERIVED_UNIT				
heat_rate_unit to derived_unit_element (as elements) (SET [1:?])	PATH		204	derived_unit derived_unit.elements[i] -> derived_unit_element
INERTIA_MOMENT_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'inertia moment measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}

Application Element	AIM element	Source	Rules	Reference Path
inertia_moment_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit { derived_unit.name = 'inertia moment unit' }
INERTIA_MOMENT_UNIT	derived_unit	41		DERIVED_UNIT_1 <'inertia moment unit', length_unit, 4.0>
- from DERIVED_UNIT				
inertia_moment_unit to derived_unit_element (as elements) (SET [1:?])	PATH		205	derived_unit derived_unit.elements[i] -> derived_unit_element
LENGTH_MEASURE_WITH_UNIT	length_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		length_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = length_measure }
length_measure_with_unit to named_unit (as unit_component)	PATH			length_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit] [length_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => length_measure_with_unit) }
LENGTH_UNIT	length_unit	41		
- from NAMED_UNIT				
length_unit to dimensional_exponents (as dimensions)	PATH			length_unit <= named_unit named_unit.dimensions -> dimensional_exponents

Application Element	AIM element	Source	Rules	Reference Path
LINEAR_STIFFNESS_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'linear stiffness measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
linear_stiffness_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'linear stiffness unit'}
LINEAR_STIFFNESS_UNIT	derived_unit	41		DERIVED_UNIT_3<'linear stiffness unit', mass_unit, 1.0, length_unit, 2.0, time_unit, -2.0>
- from DERIVED_UNIT				
linear_stiffness_unit to derived_unit_element (as elements) (SET [1:?])	PATH		206	derived_unit derived_unit.elements[i] -> derived_unit_element
LUMINOUS_INTENSITY_MEASURE_WITH_UNI T	luminous_intensity_measure_w ith_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		luminous_intensity_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = luminous_intensity_measure}
luminous_intensity_measure_with_unit to named_unit (as unit_component)	PATH			luminous_intensity_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit]

Application Element	AIM element	Source	Rules	Reference Path
				[luminous_intensity_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => luminous_intensity_measure_with_unit))
LUMINOUS_INTENSITY_UNIT	luminous_intensity_unit	41		
- from NAMED_UNIT				
luminous_intensity_unit to dimensional_exponents (as dimensions)	PATH			luminous_intensity_unit <= named_unit named_unit.dimensions -> dimensional_exponents
MASS_MEASURE_WITH_UNIT	mass_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		mass_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = mass_measure }
mass_measure_with_unit to named_unit (as unit_component)	PATH			mass_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit]) [mass_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => mass_measure_with_unit))
MASS_UNIT	mass_unit	41		
- from NAMED_UNIT				
mass_unit to dimensional_exponents (as dimensions)	PATH			mass_unit <= named_unit named_unit.dimensions -> dimensional_exponents

Application Element	AIM element	Source	Rules	Reference Path
MOMENT_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'moment measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
moment_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'moment unit'}
MOMENT_UNIT	derived_unit	41		DERIVED_UNIT_3<'moment unit', mass_unit, 1.0, length_unit, 2.0, time_unit, -2.0>
- from DERIVED_UNIT				
moment_unit to derived_unit_element (as elements) (SET [1:?])	PATH			derived_unit derived_unit.elements[i] -> derived_unit_element
NAMED_UNIT	named_unit	41		
named_unit to dimensional_exponents (as dimensions)	PATH			named_unit named_unit.dimensions -> dimensional_exponents
PLANE_ANGLE_MEASURE_WITH_UNIT	plane_angle_measure_with_uni t	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		plane_angle_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = plane_angle_measure}
plane_angle_measure_with_unit to named_unit (as unit_component)	PATH			plane_angle_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit

Application Element	AIM element	Source	Rules	Reference Path
				unit = named_unit named_unit { => ([si_unit] [plane_angle_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => plane_angle_measure_with_unit)}
PLANE_ANGLE_UNIT	plane_angle_unit	41		
- from NAMED_UNIT				
plane_angle_unit to dimensional_exponents (as dimensions)	PATH			plane_angle_unit <= named_unit named_unit.dimensions -> dimensional_exponents
PRESSURE_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'pressure measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
pressure_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'pressure unit'}
PRESSURE_UNIT	derived_unit	41		DERIVED_UNIT_3<'pressure unit', mass_unit, 1.0, length_unit, -1.0, time_unit, -2.0>
- from DERIVED_UNIT				
pressure_unit to derived_unit_element (as elements) (SET [1:?])	PATH		207	derived_unit derived_unit.elements[i] -> derived_unit_element
PRODUCT_OF_INERTIA_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'product of inertia measure with unit', 'extended measure with unit'>]}

Application Element	AIM element	Source	Rules	Reference Path
T				[ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
product_of_inertia_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'product of inertia unit'}
PRODUCT_OF_INERTIA_UNIT	derived_unit	41		DERIVED_UNIT_1<'product of inertia unit', length_unit, 5.0>
- from DERIVED_UNIT				
product_of_inertia_unit to derived_unit_element (as elements) (SET [1:?])	PATH		208	derived_unit derived_unit.elements[i] -> derived_unit_element
RATIO_MEASURE_WITH_UNIT	ratio_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		ratio_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = ratio_measure}
ratio_measure_with_unit to named_unit (as unit_component)	PATH			ratio_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit] [ratio_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => ratio_measure_with_unit)}

Application Element	AIM element	Source	Rules	Reference Path
RATIO_UNIT	ratio_unit	41		
- from NAMED_UNIT				
ratio_unit to dimensional_exponents (as dimensions)	PATH			ratio_unit <= named_unit named_unit.dimensions -> dimensional_exponents
SECTION_MODULUS_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'section modulus measure with unit', 'extended measure with unit'] [ROOT_CLASS<measure_with_unit, 'extended measure with unit']}]
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_comp onent	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
section_modulus_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'section modulus unit'}
SECTION_MODULUS_UNIT	derived_unit	41		DERIVED_UNIT_1<'section modulus unit', length_unit, 3.0>
- from DERIVED_UNIT				
section_modulus_unit to derived_unit_element (as elements) (SET [1:?])	PATH		209	derived_unit derived_unit.elements[i] -> derived_unit_element
SI_UNIT	si_unit	41		
dimensions (DER)	si_unit.dimensions	41		
name	si_unit.name	41		
prefix (OPT)	si_unit.prefix	41		
- from NAMED_UNIT				
si_unit to dimensional_exponents (as dimensions)	PATH			si_unit <= named_unit named_unit.dimensions ->

Application Element	AIM element	Source	Rules	Reference Path
				dimensional_exponents
SPEED_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'speed measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
speed_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'speed unit'}
SPEED_UNIT	derived_unit	41		DERIVED_UNIT_2<'speed unit', length_unit, 1.0, time_unit, - 1.0>
- from DERIVED_UNIT				
speed_unit to derived_unit_element (as elements) (SET [1:?])	PATH		210	derived_unit derived_unit.elements[i] -> derived_unit_element
STRESS_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'stress measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
stress_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit {derived_unit.name = 'stress unit'}

Application Element	AIM element	Source	Rules	Reference Path
STRESS_UNIT	derived_unit	41		DERIVED_UNIT_3<'stress unit', mass_unit, 1.0, length_unit, -1.0, time_unit, -2.0>
- from DERIVED_UNIT				
stress_unit to derived_unit_element (as elements) (SET [1:?])	PATH		211	derived_unit derived_unit.elements[i] -> derived_unit_element
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	thermodynamic_temperature_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		thermodynamic_temperature_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = thermodynamic_temperature_measure }
thermodynamic_temperature_measure_with_unit to named_unit (as unit_component)	PATH			thermodynamic_temperature_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit] [thermodynamic_temperature_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => thermodynamic_temperature_measure_with_unit)}
THERMODYNAMIC_TEMPERATURE_UNIT	thermodynamic_temperature_unit	41		
- from NAMED_UNIT				
thermodynamic_temperature_unit to dimensional_exponents (as dimensions)	PATH			thermodynamic_temperature_unit <= named_unit named_unit.dimensions -> dimensional_exponents
TIME_MEASURE_WITH_UNIT	time_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				

Application Element	AIM element	Source	Rules	Reference Path
value_component	measure_with_unit.value_component	41		time_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = time_measure }
time_measure_with_unit to named_unit (as unit_component)	PATH			time_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit] [time_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => time_measure_with_unit)}
TIME_UNIT	time_unit	41		
- from NAMED_UNIT				
time_unit to dimensional_exponents (as dimensions)	PATH			time_unit <= named_unit named_unit.dimensions -> dimensional_exponents
VOLTAGE_MEASURE_WITH_UNIT	measure_with_unit	41		{[CLASS<measure_with_unit, 'voltage measure with unit', 'extended measure with unit'>] [ROOT_CLASS<measure_with_unit, 'extended measure with unit'>]]}
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		{measure_with_unit.value_component -> measure_value measure_value = context_dependent_measure}
voltage_measure_with_unit to derived_unit (as unit_component)	PATH			measure_with_unit measure_with_unit.unit_component -> unit unit = derived_unit derived_unit

Application Element	AIM element	Source	Rules	Reference Path
				{derived_unit.name = 'voltage unit'}
VOLTAGE_UNIT	derived_unit	41		DERIVED_UNIT_4<'voltage unit', mass_unit, 1.0, length_unit, 2.0, time_unit, -3.0, electric_current_unit, -1.0>
- from DERIVED_UNIT				
voltage_unit to derived_unit_element (as elements) (SET [1:?])	PATH		212	derived_unit derived_unit.elements[i] -> derived_unit_element
VOLUME_MEASURE_WITH_UNIT	volume_measure_with_unit	41		
- from EXTENDED_MEASURE_WITH_UNIT				
value_component	measure_with_unit.value_component	41		volume_measure_with_unit <= measure_with_unit measure_with_unit.value_component { -> measure_value measure_value = volume_measure }
volume_measure_with_unit to named_unit (as unit_component)	PATH			volume_measure_with_unit <= measure_with_unit measure_with_unit.unit_component -> unit unit = named_unit named_unit { => ([si_unit] [volume_unit]) (conversion_based_unit conversion_based_unit.conversion_factor -> measure_with_unit => volume_measure_with_unit)}
VOLUME_UNIT	volume_unit	41		
- from NAMED_UNIT				
volume_unit to dimensional_exponents (as dimensions)	PATH			volume_unit <= named_unit named_unit.dimensions -> dimensional_exponents

Table 14 - Mapping table for structural_features UoF

Application Element	AIM element	Source	Rules	Reference Path
BEVEL_DESIGN_DEFINITION	representation	43	214, . 222, 223, 224, 225	representation {[CLASS<representation, 'bevel design definition', 'corner cutout design definition'>] [CLASS<representation, 'corner cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}
offset	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'offset', positive_length_measure>
x_y_angle	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x y angle', plane_angle_measure>
- from CORNER_CUTOUT_DESIGN_DEFINITION				
z_depth (OPT)	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
bevel_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from DESIGN_DEFINITION				
bevel_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
bevel_design_definition to global_id (as id)	<i>see supertype</i>			
bevel_design_definition to unit (as local_units)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(SET [0:?])				
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CIRCULAR_CUTOUT_DESIGN_DEFINITION	representation	43	214, . 222, 226, 227	representation {[CLASS<representation, 'circular cutout design definition', 'interior cutout design definition'] [CLASS<representation, 'interior cutout design definition', 'feature design definition'] [CLASS<representation, 'feature design definition', 'design definition'] [CLASS<representation, 'design definition', 'definition'] [CLASS<representation, 'definition', 'versionable object'] [ROOT_CLASS<representation, 'versionable object']]}
radius	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'radius', positive_length_measure>
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
circular_cutout_design_definition to feature (as defined_for) (SET [1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
circular_cutout_design_definition to representation (as representations) (SET [0:?])	<i>see supertype</i>			
- from DEFINITION				
circular_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
circular_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORNER_CUTOUT_BOUNDARY_RELATIONSHIP	extended_shape_aspect_relationship	218	215	extended_shape_aspect_relationship {[CLASS<extended_shape_aspect_relationship, 'corner cutout boundary relationship', 'structural cutout boundary relationship'>]} [CLASS<extended_shape_aspect_relationship, 'structural cutout boundary relationship', 'structural feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'structural feature relationship', 'feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'feature relationship', 'item relationship'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'definable object'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'definable object'>]]<= shape_aspect_relationship
boundary_index_2	shape_aspect	41		extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] [{shape_aspect.of_shape -> product_definition_shape <= <PROP_TO_PROD_DEF> {(CLASS_ID<product_definition, 'panel system'>) (CLASS_ID<product_definition, 'plate'>)}]}
corner_cutout_boundary_relationship to	PATH			extended_shape_aspect_relationship <=

Application Element	AIM element	Source	Rules	Reference Path
corner_cutout (as item_1) (OPT)				shape_aspect_relationship shape_aspect_relationship.relate_shape_aspect -> shape_aspect => extended_shape_aspect (CLASS_ID<extended_shape_aspect, 'corner cutout'>)
- from STRUCTURAL_CUTOUT_BOUNDARY_REL ATIONSHIP				
boundary_index	<i>see supertype</i>			
- from STRUCTURAL_FEATURE_RELATIONSHIP				
- from FEATURE_RELATIONSHIP				
- from ITEM_RELATIONSHIP				
corner_cutout_boundary_relationship to external_instance_reference (as external_item_1) (OPT)	<i>see supertype</i>			
corner_cutout_boundary_relationship to external_instance_reference (as external_item_2) (OPT)	<i>see supertype</i>			
corner_cutout_boundary_relationship to item (as item_2)	<i>see supertype</i>			
contxt (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
corner_cutout_boundary_relationship to global_id (as id)	<i>see supertype</i>			
corner_cutout_boundary_relationship to definition (as definitions) (INVERSE)	<i>see supertype</i>			
Issue: Will there ever be definitions assigned to				

Application Element	AIM element	Source	Rules	Reference Path
this thing?				
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORNER_CUTOUT_DESIGN_DEFINITION (ABS)	representation	43	. 222	<i>maps OR subtypes</i>
z_depth (OPT)	value_representation_item.v alue_component	43		representation { REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'corner cutout'>) (CLASS_ID<extended_shape_aspect, 'edge feature'>)} } representation.items [i] -> VAL_REP_ITEM<'z depth', positive_length_measure>
- from FEATURE_DESIGN_DEFINITION				
corner_cutout_design_definition to feature (as defined_for) (SET[1:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'corner cutout'>) (CLASS_ID<extended_shape_aspect, 'edge feature'>)} }
- from DESIGN_DEFINITION				
corner_cutout_design_definition to representation (as representations) (SET[0:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {(CLASS_ID<extended_shape_aspect, 'corner cutout'>) (CLASS_ID<extended_shape_aspect, 'edge feature'>)} <= shape_aspect SA2REP_D<'feature shape representation' representation

Application Element	AIM element	Source	Rules	Reference Path
				{representation.name = 'representations'} => shape_representation
- from DEFINITION				
corner_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
corner_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORNER_CUTOUT	extended_shape_aspect	218	214	extended_shape_aspect {[CLASS<extended_shape_aspect, 'corner cutout', 'structural cutout'>] [CLASS<extended_shape_aspect, 'structural cutout', 'structural feature'>] [CLASS<extended_shape_aspect, 'structural feature', 'feature'>] [CLASS<extended_shape_aspect, 'feature', 'item'>] [CLASS<extended_shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<extended_shape_aspect, 'definable object'>]]<= shape_aspect
- from STRUCTURAL_CUTOUT				
- from STRUCTURAL_FEATURE				
corner_cutout to item (as parent)	<i>see supertype</i>			
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
corner_cutout to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
description (OPT)	<i>see supertype</i>			
corner_cutout to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
corner_cutout to global_id (as id)	<i>see supertype</i>			
corner_cutout to definition (as definitions) (INVERSE) (SET[0:?])	PATH			extended_shape_aspect <= shape_aspect SA2REP_DN<'feature parameter representation', 'defined_for'> representation { CLASS_ID<representation, 'corner cutout design definition'>}
DRAIN_HOLE_CUTOUT_DESIGN_DEFINITI ON	representation	43	214, . 222, 228, 229, 230, 231, 232, 233	representation {[CLASS<representation, 'drain hole cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}
depth	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'depth', positive_length_measure>
drain_hole_radius	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'drain hole radius', positive_length_measure>
gap	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'gap', positive_length_measure>
gap_radius	value_representation_item.v alue_component	43		representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				VAL_REP_ITEM<'gap radius', positive_length_measure>
width	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'width', positive_length_measure>
drain_hole_cutout_design_definition to edge_cutout (as defined_for) (SET[1:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined for'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'edge cutout'>}
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
drain_hole_cutout_design_definition to representation (as representations) (SET[0:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'edge cutout'>} <= shape_aspect SA2REP_D<'feature shape representation'> representation {representation.name = 'representations'} => shape_representation
- from DEFINITION				
drain_hole_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
drain_hole_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
EDGE_CUTOUT	extended_shape_aspect	218	214	extended_shape_aspect

Application Element	AIM element	Source	Rules	Reference Path
				{[CLASS<extended_shape_aspect, 'edge cutout', 'structural cutout'>] [CLASS<extended_shape_aspect, 'structural cutout', 'structural feature'>] [CLASS<extended_shape_aspect, 'structural feature', 'feature'>] [CLASS<extended_shape_aspect, 'feature', 'item'>] [CLASS<extended_shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<extended_shape_aspect, 'definable object'>]}<= shape_aspect
- from STRUCTURAL_CUTOUT				
- from STRUCTURAL_FEATURE				
edge_cutout to item (as parent)	<i>see supertype</i>			
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
edge_cutout to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
edge_cutout to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
edge_cutout to global_id (as id)	<i>see supertype</i>			
edge_cutout to definition (as definitions) (INVERSE) (SET[0:?])	PATH			extended_shape_aspect <= shape_aspect shape_aspect = represented_definition represented_definition <= property_definition_representation.definition

Application Element	AIM element	Source	Rules	Reference Path
				property_definition_representation {property_definition_representation.name = definitions} property_definition_representation.used_representation -> representation {CLASS_ID<representation, 'edge cutout design definition'>}
EDGE_FEATURE	extended_shape_aspect	218	214	extended_shape_aspect {[CLASS<extended_shape_aspect, 'edge feature', 'structural cutout'>] [CLASS<extended_shape_aspect, 'structural cutout', 'structural feature'>] [CLASS<extended_shape_aspect, 'structural feature', 'feature'>] [CLASS<extended_shape_aspect, 'feature', 'item'>] [CLASS<extended_shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<extended_shape_aspect, 'definable object'>]]<= shape_aspect
- from STRUCTURAL_CUTOUT				
- from STRUCTURAL_FEATURE				
edge_feature to item (as parent)	<i>see supertype</i>			
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
edge_feature to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
edge_feature to ship (as ship_context) (OPT)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from DEFINABLE_OBJECT				
edge_feature to global_id (as id)	<i>see supertype</i>			
edge_feature to definition (as definitions) (INVERSE) (SET[0:?])	PATH			<pre> extended_shape_aspect <= shape_aspect shape_aspect = represented_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = definitions} property_definition_representation.used_representation -> representation {CLASS_ID<representation, 'corner cutout design definition'>} </pre>
EDGE_CUTOUT_FUNCTIONAL_DEFINITION	representation	41	234, 235	<pre> representation {[CLASS<representation, 'edge cutout functional definition', 'functional definition'>] [CLASS<representation, 'functional definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]} </pre>
the_function	descriptive_representation_item	45		<pre> representation representation.items [i] -> descriptive_representation_item { [descriptive_representation_item.name = 'the function'] [(descriptive_representation_item.description = 'access hole') (descriptive_representation_item.description = 'air escape') (descriptive_representation_item.description = 'air liquid escape') (descriptive_representation_item.description = 'liquid escape') (descriptive_representation_item.description = 'penetration') } </pre>

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'user defined')]
edge_cutout_functional_definition to edge_cutout (as defined_for)	PATH			representation REP2SA_DN<'feature functional representation', 'defined_for' shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'edge cutout'>}
- from FUNCTIONAL_DEFINITION				
user_def_function (OPT)	representation.description		<i>see supertype</i>	
edge_cutout_functional_definition to unit (as local_units) (SET [0:0]) Issue: How to represent the omission of something in the target field?	-		215	
- from DEFINITION				
edge_cutout_functional_definition to global_id (as id)	representation.id			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
ELLIPTICAL_CUTOUT_DESIGN_DEFINITION	representation	43	214, . 222, 236, 237, 238	representation {[CLASS<representation, 'elliptical cutout design definition', 'interior cutout design definition'>] [CLASS<representation, 'interior cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}

Application Element	AIM element	Source	Rules	Reference Path
half_axis_a	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'half axis a', positive_length_measure>
half_axis_b	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'half axis b', positive_length_measure>
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
elliptical_cutout_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
elliptical_cutout_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
elliptical_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
elliptical_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
ELONGATED_OVAL_CUTOUT_DESIGN_D EFINITION	representation	43	214, . 222, 239, 240, 241, 242	representation [CLASS<representation, 'elongated oval cutout design definition', 'interior cutout design definition'>] [CLASS<representation, 'interior cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>]

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]
distance	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'distance', positive_length_measure>
r1	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'r1', positive_length_measure>
r2	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'r2', positive_length_measure>
r3 (OPT)	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'r3', positive_length_measure>
r3_x (OPT)	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'r3 x', length_measure>
r3_y (OPT)	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'r3 y', length_measure>
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
elongated_oval_cutout_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
elongated_oval_cutout_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
elongated_oval_cutout_design_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
global_id (as id)				
elongated_oval_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
FREE_FORM_INTERIOR_CUTOUT_DESIGN_DEFINITION	representation	43	214, 243, 244	representation {[CLASS<representation, 'free form interior cutout design definition', 'interior cutout design definition'] [CLASS<representation, 'interior cutout design definition', 'feature design definition'] [CLASS<representation, 'feature design definition', 'design definition'] [CLASS<representation, 'design definition', 'definition'] [CLASS<representation, 'definition', 'versionable object'] [ROOT_CLASS<representation, 'versionable object']]}
free_form_interior_cutout_design_definition to curve (as bounding_curve)	curve	42		representation representation.items [i] -> representation_item {representation_item.name = 'bounding curve'} => geometric_representation_item => curve
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
free_form_interior_cutout_design_definition to feature (as defined_for) (SET [1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
free_form_interior_cutout_design_definition to representation (as representations) (SET [0:?])	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from DEFINITION				
free_form_interior_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
free_form_interior_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
INTERIOR_CUTOUT_DESIGN_DEFINITION	representation	43		<i>maps OR subtypes</i>
interior_cutout_design_definition to interior_cutout (as defined_for)	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'interior cutout'>}
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
interior_cutout_design_definition to representation (as representations) (SET[0:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'interior cutout'>} <= shape_aspect SA2REP_D<'feature shape representation' representation {representation.name = 'representations'} => shape_representation
- from DEFINITION				
interior_cutout_design_definition to global_id (as id)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
triangular_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
INTERIOR_CUTOUT	extended_shape_aspect	218	214	extended_shape_aspect {[CLASS<extended_shape_aspect, 'interior cutout', 'structural cutout'>] [CLASS<extended_shape_aspect, 'structural cutout', 'structural feature'>] [CLASS<extended_shape_aspect, 'structural feature', 'feature'>] [CLASS<extended_shape_aspect, 'feature', 'item'>] [CLASS<extended_shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<extended_shape_aspect, 'definable object'>]]<= shape_aspect
- from STRUCTURAL_CUTOUT				
- from STRUCTURAL_FEATURE				
interior_cutout to item (as parent)	<i>see supertype</i>			
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
interior_cutout to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
interior_cutout to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
interior_cutout to global_id (as id)	<i>see supertype</i>			
interior_cutout to definition (as definitions) (INVERSE) (SET[0:?])	PATH			<pre> extended_shape_aspect <= shape_aspect shape_aspect = represented_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = definitions} property_definition_representation.used_representation -> representation {CLASS_ID<representation, 'interior cutout design definition'>} </pre>
INTERIOR_CUTOUT_FUNCTIONAL_DEFINITION	representation	41	214, 245, 246	<pre> representation {[CLASS<representation, 'interior cutout functional definition', 'functional definition'>] [CLASS<representation, 'functional definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]} </pre>
the_function	descriptive_representation_item	45		<pre> representation representation.items [i] -> descriptive_representation_item { [descriptive_representation_item.name = 'the function'] [(descriptive_representation_item.description = 'access hole') (descriptive_representation_item.description = 'air escape') (descriptive_representation_item.description = 'air liquid escape') (descriptive_representation_item.description = 'foothold') (descriptive_representation_item.description = 'lightening hole') (descriptive_representation_item.description = 'liquid escape')] } </pre>

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'penetration') (descriptive_representation_item.description = 'user defined')]
interior_cutout_functional_definition to interior_cutout (as defined_for)	PATH			representation REP2SA_DN<'feature functional representation', 'defined_for'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'interior cutout'>}
- from FUNCTIONAL_DEFINITION				
user_def_function (OPT)	representation.description		see superty pe	
interior_cutout_functional_definition to unit (as local_units) (SET [0:0]) Issue: How to represent the omission of something in the target field?	-		215	
- from DEFINITION				
interior_cutout_functional_definition to global_id (as id)	representation.id			
- from VERSIONABLE_OBJECT				
version_id	see supertype			
INWARD_ROUND_CORNER_DESIGN_DEFINITION	representation	43	214, . 222, 247, 248, 249, 250	representation {[CLASS<representation, 'inward round corner design definition', 'round corner design definition'>] [CLASS<representation, 'round corner cutout design definition', 'corner cutout design definition'>] [CLASS<representation, 'corner cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design

Application Element	AIM element	Source	Rules	Reference Path
				definition'] [CLASS<representation, 'design definition', 'definition'] [CLASS<representation, 'definition', 'versionable object'] [ROOT_CLASS<representation, 'versionable object']}]
x_offset	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x offset', length_measure>
y_offset	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'y offset', length_measure>
- from ROUND_CORNER_DESIGN_DEFINITION				
radius	<i>see supertype</i>			
- from CORNER_CUTOUT_DESIGN_DEFINITION				
z_depth (OPT)	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
inward_round_corner_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from DESIGN_DEFINITION				
inward_round_corner_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
inward_round_corner_design_definition to global_id (as id)	<i>see supertype</i>			
inward_round_corner_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
version_id	<i>see supertype</i>			
OUTWARD_ROUND_CORNER_DESIGN_DEFINITION	representation	43	214, . 222, 251, 252, 253, 254	representation [[CLASS<representation, 'outward round corner design definition', 'round corner design definition'>] [CLASS<representation, 'round corner cutout design definition', 'corner cutout design definition'>] [CLASS<representation, 'corner cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]
x_offset	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x offset', positive_length_measure>
y_offset	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'y offset', positive_length_measure>
- from ROUND_CORNER_DESIGN_DEFINITION				
radius	<i>see supertype</i>			
- from CORNER_CUTOUT_DESIGN_DEFINITION				
z_depth (OPT)	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
outward_round_corner_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from DESIGN_DEFINITION				
outward_round_corner_design_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
representation (as representations) (SET[0:?])				
- from DEFINITION				
outward_round_corner_design_definition to global_id (as id)	<i>see supertype</i>			
outward_round_corner_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PART_EDGE_CUTOUT_DESIGN_DEFINITION	representation	43	214, . 222, . 255, 248, 257, 258, 259, 260, 261, 262, 263	representation {{[CLASS<representation, 'part edge cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}}
R_left	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'R left', positive_length_measure>
R_right	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'R right', positive_length_measure>
x_A	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x A', positive_length_measure>
x_B	value_representation_item.v alue_component	43		representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				VAL_REP_ITEM<'x B', length_measure>
x_C	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x C', length_measure>
x_D	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x D', positive_length_measure>
y_B	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'y B', length_measure>
y_C	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'y C', length_measure>
part_edge_cutout_design_definition to edge_cutout (as defined_for) (SET[1:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'edge cutout'>}
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
part_edge_cutout_design_definition to representation (as representations) (SET[0:?])	PATH			representation REP2SA_DN<'feature parameter representation', 'defined_for' shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'edge cutout'>} <= shape_aspect SA2REP_D<'feature shape representation' representation {representation.name = 'representations'} => shape_representation
- from DEFINITION				

Application Element	AIM element	Source	Rules	Reference Path
part_edge_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
part_edge_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
POSITION_FEATURE_DESIGN_DEFINITION (ABS)	representation	43		<i>maps OR subtypes</i>
position_feature_design_definition to shape_representation (as representations)	PATH			<pre> representation REP2SA_DN<'feature parameter representation', 'defined_for'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'position feature'>} <= shape_aspect SA2REP_D<'feature shape representation'> representation {representation.name = 'representations'} => shape_representation </pre>
position_feature_design_definition to position_feature (as defined_for)	PATH			<pre> representation REP2SA_DN<'feature parameter representation', 'defined_for'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'position feature'>} </pre>
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
- from DEFINITION				
position_feature_design_definition to global_id	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(as id)				
position_feature_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
POSITION_FEATURE (ABS)	<i>see subtypes</i>			
- from STRUCTURAL_FEATURE				
position_feature to item (as parent)	<i>see supertype</i>			
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
position_feature to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
position_feature to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
position_feature to global_id (as id)	<i>see supertype</i>			
position_feature to definition (as definitions) (INVERSE) (SET[0:?])	<i>see subtypes</i>			
POSITION_FEATURE_RELATIONSHIP (ABS)	<i>see subtypes</i>			
position_feature_relationship to position_feature (as item_1)	<i>see subtypes</i>			
RECTANGULAR_CUTBACK_CORNER_DES IGN_DEFINITION	representation	43	214, . 222, 264,	representation {[CLASS<representation, 'rectangular cutback corner design definition', 'corner cutout design definition']}]

Application Element	AIM element	Source	Rules	Reference Path
			265, 266, 267	[CLASS<representation, 'corner cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]
x_depth	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x depth', positive_length_measure>
y_depth	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'y depth', positive_length_measure>
radius	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'radius', positive_length_measure>
- from CORNER_CUTOUT_DESIGN_DEFINITION				
z_depth (OPT)	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
rectangular_cutback_corner_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from DESIGN_DEFINITION				
rectangular_cutback_corner_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
rectangular_cutback_corner_design_definition to global_id (as id)	<i>see supertype</i>			
rectangular_cutback_corner_design_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
unit (as local_units) (SET [0:?])				
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
RECTANGULAR_CUTOUT_DESIGN_DEFINITION	representation	43	214, . 222, 268, 269, 270	representation {[CLASS<representation, 'rectangular cutout design definition', 'interior cutout design definition'>] [CLASS<representation, 'interior cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}
length_of	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'length of', positive_length_measure>
width	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'width', positive_length_measure>
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
rectangular_cutout_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
rectangular_cutout_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
rectangular_cutout_design_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
global_id (as id)				
rectangular_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
ROUND_CORNER_DESIGN_DEFINITION (ABS)	representation	43		<i>maps OR subtypes</i>
radius	positive_length_measure	41		representation representation.items [i] -> VAL_REP_ITEM<'radius', positive_length_measure>
- from CORNER_CUTOUT_DESIGN_DEFINITION				
z_depth (OPT)	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
round_corner_design_definition to feature (as defined_for) (SET [1:?])	<i>see supertype</i>			
- from DESIGN_DEFINITION				
round_corner_design_definition to representation (as representations) (SET [0:?])	<i>see supertype</i>			
- from DEFINITION				
round_corner_design_definition to global_id (as id)	<i>see supertype</i>			
round_corner_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
ROUND_CORNER_RECTANGULAR_CUTO	representation	43	214, .	representation

Application Element	AIM element	Source	Rules	Reference Path
UT_DESIGN_DEFINITION			222, 271, 272, 273, 274	{[CLASS<representation, 'round corner rectangular cutout design definition', 'rectangular cutout design definition'] [CLASS<representation, 'rectangular cutout design definition', 'interior cutout design definition'] [CLASS<representation, 'interior cutout design definition', 'feature design definition'] [CLASS<representation, 'feature design definition', 'design definition'] [CLASS<representation, 'design definition', 'definition'] [CLASS<representation, 'definition', 'versionable object'] [ROOT_CLASS<representation, 'versionable object']]}
corner_radius	value_representation_item.value_component	43		representation representation.items [i] -> VAL_REP_ITEM<'corner radius', positive_length_measure>
- from RECTANGULAR_CUTOUT_DESIGN_DEFINITION				
length_of	<i>see supertype</i>			
width	<i>see supertype</i>			
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
rectangular_cutout_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
rectangular_cutout_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
rectangular_cutout_design_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
global_id (as id)				
rectangular_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
ROUND_EDGE_RECTANGULAR_CUTOUT_DESIGN_DEFINITION	representation	43	214, . 222, 275, 276, 277, 278, 279	representation {[CLASS<representation, 'round edge rectangular cutout design definition', 'rectangular cutout design definition'] [CLASS<representation, 'rectangular cutout design definition', 'interior cutout design definition'] [CLASS<representation, 'interior cutout design definition', 'feature design definition'] [CLASS<representation, 'feature design definition', 'design definition'] [CLASS<representation, 'design definition', 'definition'] [CLASS<representation, 'definition', 'versionable object'] [ROOT_CLASS<representation, 'versionable object']]}
edge_radius	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'edge radius', positive_length_measure>
distance	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'distance', length_measure>
- from RECTANGULAR_CUTOUT_DESIGN_DEFIN ITION				
length_of	<i>see supertype</i>			
width	<i>see supertype</i>			
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				

Application Element	AIM element	Source	Rules	Reference Path
rectangular_cutout_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
rectangular_cutout_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
rectangular_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
rectangular_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
SEAM_CURVE_RELATIONSHIP	extended_shape_aspect_relationship	218	214, . 222, 280, 281	extended_shape_aspect_relationship {[CLASS<extended_shape_aspect_relationship, 'seam curve relationship', 'position feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'position feature relationship', 'structural feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'structural feature relationship', 'feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'feature relationship', 'item relationship'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'definable object'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'definable object'>]]<=

Application Element	AIM element	Source	Rules	Reference Path
boundary_index (OPT)	shape_aspect	41		shape_aspect_relationship extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] [{shape_aspect.of_shape -> product_definition_shape <= PROP_TO_PROD_DEF {CLASS_ID<product_definition, 'panel system'>} }]
displacement (OPT)	value_representation_item.v alue_component	43	216	extended_shape_aspect_relationship <= shape_aspect_relationship represented_definition = shape_aspect_relationship represented_definition <= PDR<'seam curve relationship parameter'> -> representation representation.items [i] -> VAL_REP_ITEM<'displacement', length_measure>
seam_curve_relationship to seam or panel_system (as item_2) #1: to seam #2: to panel_system	PATH			#1: extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'seam'>} #2: extended_shape_aspect_relationship <= shape_aspect_relationship <= shape_aspect_relationship.related_shape_aspect -> shape_aspect {CLASS_ID<shape_aspect, 'border'>} shape_aspect.of_shape -> product_definition_shape <= <PROP_TO_PROD_DEF>

Application Element	AIM element	Source	Rules	Reference Path
				{CLASS_ID<product_definition, 'panel system'>}
- from POSITION_FEATURE_RELATIONSHIP				
seam_curve_relationship to position_feature (as item_1)	PATH			extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.relate_shape_aspect -> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'seam'>}
- from STRUCTURAL_FEATURE_RELATIONSHIP				
- from FEATURE_RELATIONSHIP				
- from ITEM_RELATIONSHIP				
contxt (OPT)	extended_shape_aspect_relationship.description	218		
seam_curve_relationship to external_instance_reference (as external_item_1) (OPT) Issue: no ARM constraint on type of external instance. Issue: Where shall the attribute/assertion name go?	PATH			extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'extended shape aspect'>
seam_curve_relationship to external_instance_reference (as external_item_2) (OPT) #1: to seam #2: to panel_system	PATH			#1: (extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'extended shape aspect'> {CLASS_ID<extended_shape_aspect, 'seam'>}) #2: (extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'product definition'>

Application Element	AIM element	Source	Rules	Reference Path
				{CLASS_ID<product_definition, 'panel system'>})
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.as signed_id	41	221	extended_shape_aspect_relationship <- HAS_ID<group, 'version id'> identification_assignment.assigned_id
- from DEFINABLE_OBJECT				
seam_curve_relationship to global_id (as id)	extended_shape_aspect_rela tionship.id	218		
seam_curve_relationship to definition (as definitions) (INV)	???			
SEAM_DESIGN_DEFINITION	representation	43		representation [[CLASS<representation, 'seam design definition', 'position feature design definition'>] [CLASS<representation, 'position feature design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]
seam_curve #1: if the seam curve is a curve #2: if the seam curve is a seam curve relationship	#1: (curve) #2: (extended_shape_aspect_rel ationship)	42 43	#1: 217 #2: 219	#1: (representation representation.items [i] -> representation_item {representation_item.name = 'seam curve'} => geometric_representation_item => curve) #2: (representation REP2SA_DN<'UNUSED.', 'seam curve'> shape_aspect {(=> extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'seam'>})

Application Element	AIM element	Source	Rules	Reference Path
				<pre> ({CLASS_ID<shape_aspect, 'border'>} shape_aspect.of_shape -> product_definition_shape <= <PROP_TO_PROD_DEF> {CLASS_ID<product_definition, 'panel system'>}}) <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => extended_shape_aspect_relationship {CLASS_ID<extended_shape_aspect_relationship, 'seam curve relationship'>} </pre>
seam_design_definition to seam (as border) (LIST [0:2])	PATH		218	<pre> representation <- REP2SA_DN<'.UNUSED.', 'border'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'seam'>} </pre>
seam_design_definition to seam (as defined_for)	PATH			<pre> representation REP2SA_DN<'feature parameter representation', 'defined_for'> shape_aspect => extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'seam'>} </pre>
- from POSITION_FEATURE_DESIGN_DEFINITION				
seam_design_definition to shape_representation (as representations)	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
- from DEFINITION				
seam_design_definition to global_id (as id)	<i>see supertype</i>			
seam_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
version_id	<i>see supertype</i>			
SEAM	extended_shape_aspect	218	214	extended_shape_aspect {[CLASS<extended_shape_aspect, 'seam', 'position feature'>]} [CLASS<extended_shape_aspect, 'position feature', 'structural feature'>]} [CLASS<extended_shape_aspect, 'structural feature', 'feature'>]} [CLASS<extended_shape_aspect, 'feature', 'item'>]} [CLASS<extended_shape_aspect, 'item', 'definable object'>]} [ROOT_CLASS<extended_shape_aspect, 'definable object'>]}<= shape_aspect
seam to panel_system (as parent)	PATH			extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'panel system'>}
- from POSITION_FEATURE				
- from STRUCTURAL_FEATURE				
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
seam to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(OPT)				
seam to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
seam to global_id (as id)	<i>see supertype</i>			
seam to definition (as definitions) (INVERSE) (SET[0:?])	PATH			extended_shape_aspect <= shape_aspect SA2REP_DN<'feature parameter representation', 'defined_for' representation {CLASS_ID<representation, 'seam design definition'>}
SHEAR_BEVEL_DESIGN_DEFINITION	representation	43	214, . 222, . 282, 283, 284, 285	representation {[CLASS<representation, 'shear bevel design definition', 'bevel design definition'>] [CLASS<representation, 'bevel design definition', 'corner cutout design definition'>] [CLASS<representation, 'corner cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}
x_z_angle	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'x z angle', positive_angle_measure>
- from BEVEL_DESIGN_DEFINITION				
offset	<i>see supertype</i>			
x_y_angle	<i>see supertype</i>			
- from CORNER_CUTOUT_DESIGN_DEFINITION				
z_depth	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(OPT)				
- from FEATURE_DESIGN_DEFINITION				
shear_bevel_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from DESIGN_DEFINITION				
shear_bevel_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
shear_bevel_design_definition to global_id (as id)	<i>see supertype</i>			
shear_bevel_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
STRUCTURAL_CUTOUT_BOUNDARY_REL ATIONSHIP	extended_shape_aspect_rela tionship	218		extended_shape_aspect_relationship {[CLASS<extended_shape_aspect_relationship, 'structural cutout boundary relationship', 'structural feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'structural feature relationship', 'feature relationship'>] [CLASS<extended_shape_aspect_relationship, 'feature relationship', 'item relationship'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'definable object'>] [CLASS<extended_shape_aspect_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship,

Application Element	AIM element	Source	Rules	Reference Path
				'definable object'>]]<= shape_aspect_relationship
boundary_index	shape_aspect	41		extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect [{CLASS_ID<shape_aspect, 'border'>}] [{shape_aspect.of_shape -> product_definition_shape <= <PROP_TO_PROD_DEF> {(CLASS_ID<product_definition, 'panel system'>) (CLASS_ID<product_definition, 'plate'>)}]
structural_cutout_boundary_relationship to structural_cutout (as item_1)	PATH			extended_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.relate_shape_aspect -> shape_aspect => extended_shape_aspect (CLASS_ID<extended_shape_aspect, 'structural cutout'>)
- from STRUCTURAL_FEATURE_RELATIONSHIP				
- from FEATURE_RELATIONSHIP				
- from ITEM_RELATIONSHIP				
structural_cutout_boundary_relationship to external_instance_reference (as external_item_1) (OPT)	PATH			extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'extended shape aspect'>
structural_cutout_boundary_relationship to external_instance_reference (as external_item_2) (OPT)	PATH			extended_shape_aspect_relationship EXT_INST_REF<extended_shape_aspect_relationship, 'ship structures', 'product definition'> {(CLASS_ID<product_definition, 'plate'>) (CLASS_ID<product_definition, 'panel system'>)}
structural_cutout_boundary_relationship to item (as item_2)	PATH			extended_shape_aspect_relationship <= shape_aspect_relationship <-

Application Element	AIM element	Source	Rules	Reference Path
				shape_aspect_relationship.related_shape_aspect -> shape_aspect [{{CLASS_ID<shape_aspect, 'border'}}] shape_aspect.of_shape -> product_definition_shape <= PROP_TO_PROD_DEF {{(CLASS_ID<product_definition, 'panel system' >)} (CLASS_ID<product_definition, 'plate' >)}
contxt (OPT)	extended_shape_aspect_relationship.description	218		
- from DEFINABLE_OBJECT				
structural_cutout_boundary_relationship to global_id (as id)	extended_shape_aspect_relationship.id	218		
structural_cutout_boundary_relationship to definition (as definitions) (INVERSE)	???			
Issue: Will there ever be definitions assigned to this thing?				
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.asigned_id	41	221	extended_shape_aspect_relationship HAS_ID<extended_shape_aspect_relationship, 'version id'> identification_assignment.assigned_id
STRUCTURAL_CUTOUT (ABS)	extended_shape_aspect			<i>see subtypes</i>
structural_cutout to item (as parent)	<i>see supertype</i>			
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
structural_cutout to external_reference (as documentation)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(SET[0:?])				
description (OPT)	<i>see supertype</i>			
structural_cutout to ship (as ship_context) (OPT)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
structural_cutout to global_id (as id)	<i>see supertype</i>			
structural_cutout to definition (as definitions) (INVERSE) (SET[0:?])	<i>see subtypes</i>			
STRUCTURAL_FEATURE (ABS)	<i>see subtypes</i>			
structural_feature to item (as parent)	PATH		220	extended_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'parent'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'item'>}
- from FEATURE				
- from ITEM				
name	<i>see supertype</i>			
structural_feature to external_reference (as documentation) (SET[0:?])	<i>see supertype</i>			
description (OPT)	<i>see supertype</i>			
structural_feature to ship (as ship_context)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
(OPT)				
- from DEFINABLE_OBJECT				
structural_feature to global_id (as id)	<i>see supertype</i>			
structural_feature to definition (as definitions) (INVERSE) (SET[0:?])	<i>see subtypes</i>			
STRUCTURAL_FEATURE_RELATIONSHIP (ABS)	<i>see subtypes</i>			
structural_feature_relationship to structural_feature (as item_1)	<i>see subtypes</i>			
- from FEATURE_RELATIONSHIP				
- from ITEM_RELATIONSHIP				
v to external_instance_reference (as external_item_1) (OPT)	<i>see subtypes</i>			
structural_feature_relationship to external_instance_reference (as external_item_2) (OPT)	<i>see subtypes</i>			
structural_feature_relationship to item (as item_2)	<i>see subtypes</i>			
contxt	<i>see subtypes</i>			
- from DEFINABLE_OBJECT				
structural_feature_relationship to global_id (as id)	<i>see subtypes</i>			
structural_feature_relationship to definition (as definitions) (INVERSE)	<i>see subtypes</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see subtypes</i>			
STRUCTURAL_PART_PENETRATION_REL ATIONSHIP	product_definition_relations hip	41		product_definition_relationship {[CLASS<product_definition_relationship, 'structural part

Application Element	AIM element	Source	Rules	Reference Path
				penetration relationship', 'item relationship'] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'definable object'>]}
structural_part_penetration_relationship to structural_cutout as (penetration_result) (SET [1:?])	PATH			extended_shape_aspect {CLASS_ID<extended_shape_aspect, 'structural cutout'} <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.name = 'penetration result'} property_definition.definition -> characterized_definition = characterized_product_definition = product_definition_relationship
structural_part_penetration_relationship to structural_part (as item_1)	PATH			product_definition_relationship product_definition_relationship.relateing_product_definitio n -> product_definition {CLASS_ID<product_definition, 'structural part'>}
- from ITEM_RELATIONSHIP				
structural_part_penetration_relationship to external_instance_reference (as external_item_1) (OPT)	PATH			product_definition_relationship EXT_INST_REF<product_definition_relationship, 'ship structures', 'product definition'>
structural_part_penetration_relationship to external_instance_reference (as	PATH to Part or System			product_definition_relationship EXT_INST_REF<product_definition_relationship,

Application Element	AIM element	Source	Rules	Reference Path
external_item_2) (OPT) Issue: How to restrict the Parts and Systems to be something useful?				'ship structures', 'product definition'>
structural_part_penetration_relationship to item (as item_2)	PATH			product_definition_relationship product_definition_relationship.relying_product_definitio n -> product_definition {(CLASS_ID<product_definition, 'part'>) (CLASS_ID<product_definition, 'system'>)}
contxt	product_definition_relations hip.description	41		
- from DEFINABLE_OBJECT				
structural_part_penetration_relationship to global_id (as id)	product_definition_relations hip.id	41		
structural_part_penetration_relationship to definition (as definitions) (INVERSE) Issue: Will there ever be a Definition that points to a this relationship?	???			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.as signed_id	41	221	product_definition_relationship HAS_ID< product_definition_relationship, 'version id'> identification_assignment.assigned_id
STRUCTURAL_SYSTEM_PENETRATION_R ELATIONSHIP	product_definition_relations hip	41		product_definition_relationship {[CLASS<product_definition_relationship, 'structural system penetration relationship', 'item relationship'] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[ROOT_CLASS<product_definition_relationship, 'versionable object'>] [ROOT_CLASS<extended_shape_aspect_relationship, 'definable object'>]]
structural_system_penetration_relationship to structural_cutout (as penetration_result) (SET [1:?])	PATH			extended_shape_aspect { CLASS_ID<extended_shape_aspect, 'structural cutout' =<= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition { property_definition.name = 'penetration result' property_definition.definition -> characterized_definition = characterized_product_definition = product_definition_relationship
structural_system_penetration_relationship to structural_system (as item_1)	PATH			product_definition_relationship product_definition_relationship.relating_product_definitio n -> product_definition { CLASS_ID<product_definition, 'structural system'}
- from ITEM_RELATIONSHIP				
structural_system_penetration_relationship to external_instance_reference (as external_item_1) (OPT)	PATH			product_definition_relationship EXT_INST_REF<product_definition_relationship, 'ship structures', 'product definition'>
structural_system_penetration_relationship to external_instance_reference (as external_item_2) (OPT)	PATH to Part or System			product_definition_relationship EXT_INST_REF<product_definition_relationship, 'ship structures', 'product definition'>
Issue: How to restrict the Parts and Systems (product_definitions) to be something useful?				

Application Element	AIM element	Source	Rules	Reference Path
structural_system_penetration_relationship to item (as item_2)	PATH			product_definition_relationship product_definition_relationship.relateing_product_definitio n -> product_definition {(CLASS_ID<product_definition, 'part'>) (CLASS_ID<product_definition, 'system'>)}
contxt	product_definition_relations hip.description	41		
- from DEFINABLE_OBJECT				
structural_system_penetration_relationship to global_id (as id)	product_definition_relations hip.id	41		
structural_system_penetration_relationship to definition (as definitions) (INVERSE) Issue: Will there ever be a Definition that points to a this relationship?	???			
- from VERSIONABLE_OBJECT				
version_id	identification_assignment.as signed_id	41	221	product_definition_relationship HAS_ID< product_definition_relationship, 'version id'> identification_assignment.assigned_id
TRIANGULAR_CUTOUT_DESIGN_DEFINIT ION	representation	43	214, . 222, 286, 287, 288, 289, 290, 291, 292	representation {[CLASS<representation, 'triangular cutout design definition', 'interior cutout design definition'>] [CLASS<representation, 'interior cutout design definition', 'feature design definition'>] [CLASS<representation, 'feature design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]]}
c2_x	value_representation_item.v alue_component	43		representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				VAL_REP_ITEM<'c2 x', positive_length_measure>
c3_x	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'c3 x', length_measure>
c3_y	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'c3 y', positive_length_measure>
c1_radius	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'c1 radius', positive_length_measure>
c2_radius	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'c2 radius', positive_length_measure>
c3_radius	value_representation_item.v alue_component	43		representation representation.items [i] -> VAL_REP_ITEM<'c3 radius', positive_length_measure>
- from INTERIOR_CUTOUT_DESIGN_DEFINITION				
triangular_cutout_design_definition to feature (as defined_for) (SET[1:?])	<i>see supertype</i>			
- from FEATURE_DESIGN_DEFINITION				
- from DESIGN_DEFINITION				
triangular_cutout_design_definition to representation (as representations) (SET[0:?])	<i>see supertype</i>			
- from DEFINITION				
triangular_cutout_design_definition to global_id (as id)	<i>see supertype</i>			
triangular_cutout_design_definition to unit (as local_units) (SET [0:?])	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			

Table 15 - Mapping table for structural_parts UoF

Application Element	AIM element	Source	Rules	Reference Path
ANGLE_BAR_CROSS_SECTION	shape_aspect	41	331	{[CLASS<shape_aspect, 'angle bar cross section', 'angle profile cross section'] [CLASS<shape_aspect, 'angle profile cross section', 'parametric profile cross section'] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'] [ROOT_CLASS<shape_aspect, 'profile cross section']}]}
depth	value_representation_item. value_component	43	294, 301	shape_aspect = represented_definition <- PDR<'angle bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 301	shape_aspect = represented_definition <- PDR<'angle bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
radius	value_representation_item. value_component	43	294, 301	shape_aspect = represented_definition <- PDR<'angle bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius', positive_length_measure>
thk	value_representation_item. value_component	43	294, 301	shape_aspect = represented_definition <- PDR<'angle bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'thk', positive_length_measure>

Application Element	AIM element	Source	Rules	Reference Path
- from PROFILE_CROSS_SECTION				
angle_bar_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
ANGLE_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: angle_bar_cross_section				
#2: t_bar_cross_section				
- from PROFILE_CROSS_SECTION				
angle_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
BAR_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: bulb_flat_cross_section				
#2: flat_bar_cross_section				
#3: round_bar_cross_section				
- from PROFILE_CROSS_SECTION				
bar_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
BULBFLAT_CROSS_SECTION	shape_aspect	41	332	{[CLASS<shape_aspect, 'bulbflat cross section', 'bar profile cross section'>] [CLASS<shape_aspect, 'bar profile cross section', 'parametric profile cross section'>] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>]]}
depth	value_representation_item.	43	294,	shape_aspect =

Application Element	AIM element	Source	Rules	Reference Path
	value_component		302	<pre> represented_definition <- PDR<'bulb flat cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure> </pre>
bulb_width	value_representation_item. value_component	43	294, 302	<pre> shape_aspect = represented_definition <- PDR<'bulb flat cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'bulb width', positive_length_measure> </pre>
bulb_radius	value_representation_item. value_component	43	294, 302	<pre> shape_aspect = represented_definition <- PDR<'bulb flat cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'bulb radius', positive_length_measure> </pre>
tw	value_representation_item. value_component	43	294, 302	<pre> shape_aspect = represented_definition <- PDR<'bulb flat cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'tw', positive_length_measure> </pre>
- from PROFILE_CROSS_SECTION				
bulbflat_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
CHANNEL_CROSS_SECTION	shape_aspect	41	333	<pre> {[CLASS<shape_aspect, 'channel cross section', 'channel profile cross section'>]} [CLASS<shape_aspect, 'channel profile cross section', 'parametric profile cross section'>] </pre>

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]]
depth	value_representation_item. value_component	43	294, 303	shape_aspect = represented_definition <- PDR<'channel cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 303	shape_aspect = represented_definition <- PDR<'channel cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
web_thk	value_representation_item. value_component	43	294, 303	shape_aspect = represented_definition <- PDR<'channel cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'web thk', positive_length_measure>
flange_thk	value_representation_item. value_component	43	294, 303	shape_aspect = represented_definition <- PDR<'channel cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'flange thk', positive_length_measure>
radius	value_representation_item. value_component	43	294, 303	shape_aspect = represented_definition <- PDR<'channel cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius', positive_length_measure>

Application Element	AIM element	Source	Rules	Reference Path
k	value_representation_item. value_component	43	294, 303	shape_aspect = represented_definition <- PDR<'channel cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'k', positive_length_measure>
- from PROFILE_CROSS_SECTION				
channel_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
CHANNEL_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: channel_cross_section				
- from PROFILE_CROSS_SECTION				
channel_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
CIRCULAR_HOLLOW_PROFILE_CROSS_SECTION	shape_aspect	41	334	{[CLASS<shape_aspect, 'circular hollow profile cross section', 'parametric profile cross section'>] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]]}
outer_diameter	value_representation_item. value_component	43	294, 304	shape_aspect = represented_definition <- PDR<'circular hollow profile cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'outer diameter', positive_length_measure>
inner_diameter	value_representation_item. value_component	43	294, 304	shape_aspect = represented_definition <-

Application Element	AIM element	Source	Rules	Reference Path
				PDR<'circular hollow profile cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'inner diameter', positive_length_measure>
- from PROFILE_CROSS_SECTION				
circular_hollow_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
CORRUGATED_PART	product_definition	41		{[CLASS<product_definition, 'corrugated_part', 'structural part'>] [CLASS<product_definition, 'structural part', 'part'>] [CLASS<product_definition, 'part', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]]}
- from ITEM				
name	<i>see supertype</i>			
corrugated_part to external_reference (as documentation)	<i>see supertype</i>			
description	<i>see supertype</i>			
corrugated_part to ship (as ship_context)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
corrugated_part to global_id (as id)	<i>see supertype</i>			
corrugated_part to definition (as definitions)	<i>see supertype</i>			
CORRUGATED_PART_DESIGN_DEFINITION	extended_product_definition_shape	218	327	{[CLASS<extended_product_definition_shape, 'corrugated part design definition', 'structural part design definition'>] [CLASS<extended_product_definition_shape, 'structural part design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition',

Application Element	AIM element	Source	Rules	Reference Path
				'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
corrugated_part_design_definition to corrugated_part_boundary (as border) #1: border is a surface #2: border is a structural_system_adjacency_relationship #3: border is a corrugated_structure_boundary_relationship	#1: PATH #2: PATH #3: PATH		326	#1: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect = represented_definition <= PDR<'border representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', surface>]]} #2: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <= shape_aspect_relationship.relatng_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'structural system adjacency relationship'>}}} #3: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <= shape_aspect_relationship.relatng_shape_aspect

Application Element	AIM element	Source	Rules	Reference Path
				shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'corrugated structure boundary relationship'>}}
mirrored	descriptive_representation_item. description	45	305	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'corrugated part design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'mirrored'> {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
offset	value_representation_item. value_component	43	294, 305	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'corrugated part design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'offset', length_measure>
repetition	value_representation_item. value_component	43	294, 305	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'corrugated part design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'repetition', count_measure>
thickness	value_representation_item. value_component	43	294, 305	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'corrugated part design parameters'> ->

Application Element	AIM element	Source	Rules	Reference Path
				representation representation.items[I] -> VAL_REP_ITEM<'thickness', positive_length_measure>
corrugated_part_design_definition to corrugation (as shape_aspect)	PATH		328	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {CLASS_ID<shape_aspect, 'corrugation'>}
corrugated_part_design_definition to corrugated_part (as defined_for)	PATH		!only S[1:1] possible!	extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'corrugated part'>}
- from STRUCTURAL_PART_DESIGN_DEFINITION				
mass	value_representation_item. value_component	43	294, 306	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'corrugated part design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'mass', mass_measure>
corrugated_part_design_definition to shape_representation (as representations)	<i>see supertype</i>			
- from DEFINITION				
corrugated_part_design_definition to global_id (as id)	<i>see supertype</i>			
corrugated_part_design_definition to unit (as local_units)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORRUGATION	shape_aspect	41	329	{[CLASS<shape_aspect, 'corrugation', 'structural part'>] [CLASS<shape_aspect, 'structural part', 'part'>] [CLASS<shape_aspect, 'part', 'item'>] [CLASS<shape_aspect, 'item', 'definable object'>] [ROOT_CLASS<shape_aspect, 'definable object'>]}
depth	value_representation_item. value_component	43	294, 307	shape_aspect = represented_definition <- PDR<'corrugation design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
flat_width_1	value_representation_item. value_component	43	294, 307	shape_aspect = represented_definition <- PDR<'corrugation design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'flat width 1', positive_length_measure>
flat_width_2	value_representation_item. value_component	43	294, 307	shape_aspect = represented_definition <- PDR<'corrugation design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'flat width 2', positive_length_measure>
radius_1	value_representation_item. value_component	43	294, 308	shape_aspect = represented_definition <- PDR<'corrugation design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius 1', positive_length_measure>

Application Element	AIM element	Source	Rules	Reference Path
radius_2	value_representation_item. value_component	43	294, 308	shape_aspect = represented_definition <- PDR<'corrugation design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius 2', positive_length_measure>
slope_width	value_representation_item. value_component	43	294, 307	shape_aspect = represented_definition <- PDR<'corrugation design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'slope width', positive_length_measure>
EXPLICIT_PROFILE_CROSS_SECTION	shape_aspect	41	335	{[CLASS<shape_aspect, 'explicit profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]}
explicit_profile_cross_section to bounded_curve (as cross_section_geometry)	PATH		294, 309	Shape_aspect = represented_definition <- PDR<'explicit profile cross section shape'> -> representation => representation.items[i] -> GEO_REP_ITEM<'cross section geometry', bounded_curve>
explicit_profile_cross_section to axis2_placement_2d (as local_coordinate_system)	PATH		294, 309	shape_aspect = represented_definition <- PDR<'explicit profile cross section shape'> -> representation => representation.items[i] -> GEO_REP_ITEM<'local coordinate system', axis2_placement_2d>
- from PROFILE_CROSS_SECTION				
expilicite_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
designation	<i>see supertype</i>			
FLANGED_PLATE_CROSS_SECTION	shape_aspect	41	337	{[CLASS<shape_aspect, 'flanged profile cross section', 'proprietary profile cross section'] [CLASS<shape_aspect, 'proprietary profile cross section', 'parametric profile cross section'] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'] [ROOT_CLASS<shape_aspect, 'profile cross section']}]
depth	value_representation_item. value_component	43	294, 310	shape_aspect = represented_definition <- PDR<'flanged plate cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 310	shape_aspect = represented_definition <- PDR<'flanged plate cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
radius	value_representation_item. value_component	43	294, 310	shape_aspect = represented_definition <- PDR<'flanged plate cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius', positive_length_measure>
thk	value_representation_item. value_component	43	294, 310	shape_aspect = represented_definition <- PDR<'flanged plate cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'thk', positive_length_measure>
- from PROFILE_CROSS_SECTION				

Application Element	AIM element	Source	Rules	Reference Path
flanged_plate_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
FLANGED_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: w_shape_cross_section - from PROFILE_CROSS_SECTION				
flanged_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
FLAT_BAR_CROSS_SECTION	shape_aspect	41	336	{[CLASS<shape_aspect, 'flat bar cross section', 'bar profile cross section'>] [CLASS<shape_aspect, 'bar profile cross section', 'parametric profile cross section'>] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]}
depth	value_representation_item. value_component	43	294, 311	shape_aspect = represented_definition <- PDR<'flat bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 311	shape_aspect = represented_definition <- PDR<'flat bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
- from PROFILE_CROSS_SECTION				
flat_bar_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
LIBRARY_PROFILE_CROSS_SECTION	[shape_aspect]	41		{[CLASS<shape_aspect, 'library profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]}
library_profile_cross_section to library_element_reference (as library_reference)	PATH		330	shape_aspect = classification_item <- applied_classification_assignment.items applied_classification_assignment. {applied_classification_assignment.role = 'definitional class membership'}
- from PROFILE_CROSS_SECTION				
library_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
NON_CIRCULAR_HOLLOW_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: square_tube_cross_section				
- from PROFILE_CROSS_SECTION				
non_circular_hollow_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
PARAMETRIC_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: angle_profile_cross_section				
#2: bar_profile_cross_section				
#3: channel_profile_cross_section				
#4: circular_hollow_profile_cross_section				
#5: flanged_profile_cross_section				

Application Element	AIM element	Source	Rules	Reference Path
#6: non_circular_hollow_profile_cross_section				
#7: proprietary_profile_cross_section				
- from PROFILE_CROSS_SECTION				
parametric_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
PLATE	product_definition	41		{[CLASS<product_definition, 'plate', 'structural part'>] [CLASS<product_definition, 'structural part', 'part'> [CLASS<product_definition, 'part', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]}
- from ITEM				
name	<i>see supertype</i>			
plate to external_reference (as documentation)	<i>see supertype</i>			
description	<i>see supertype</i>			
plate to ship (as ship_context)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
plate to global_id (as id)	<i>see supertype</i>			
plate to definition (as definitions) (INV)	<i>see supertype</i>			
PLATE_BOUNDARY_RELATIONSHIP	product_definition_relations hip	41		{[CLASS<product_definition_relationship, 'plate boundary relationship', 'plate relationship'>] [CLASS<product_definition_relationship, 'plate relationship', 'structural part relationship'>] [CLASS<product_definition_relationship, 'structural part relationship', 'part relationship'>] [CLASS<product_definition_relationship, 'part relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]
plate_boundary_relationship to seam (as item_2)	PATH			product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape <- shape_aspect <- {(CLASS_ID<shape_aspect, 'seam'>) (shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'plate boundary relationship'>})}}
- from PLATE_RELATIONSHIP				
plate_boundary_relationship to plate (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				
plate_boundary_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
plate_boundary_relationship to external_instance_reference (as external_item_2)	PATH			product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape <- shape_aspect {[CLASS_ID<shape_aspect, 'seam'>] [EXT_INST_REF<shape_aspect, 'ship structures',

Application Element	AIM element	Source	Rules	Reference Path
				'seam'>]]}
contxt	<i>See supertype</i>		here	
- from DEFINABLE_OBJECT				
plate_boundary_relationship to global_id (as id)	<i>see supertype</i>			
plate_boundary_relationship to definition (as definitions) (INV)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PLATE_DESIGN_DEFINITION	extended_product_definition_shape	218	339	{[CLASS<extended_product_definition_shape, 'plate design definition', 'structural part design definition'>] [CLASS<extended_product_definition_shape, 'structural part design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
plate_design_definition to plate_boundary (as border) #1: border is a bounded_curve #2: border is a panel_system_boundary #3: border is a plate_boundary_relationship	#1: PATH #2: PATH #3: PATH		338	#1: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect = represented_definition <= PDR<'border representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', bounded_curve>]]} #2: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape

Application Element	AIM element	Source	Rules	Reference Path
				<pre> shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <- shape_aspect_relationship.relate_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {CLASS_ID<shape_aspect, 'border'>} shape_aspect.of_shape -> extended_product_definition_shape {CLASS_ID<extended_product_definition_shape, 'panel system design definition'>}}]} #3: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <- shape_aspect_relationship.relate_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'plate boundary relationship'>}}]} </pre>
material_offset	value_representation_item. value_component	43	294, 312	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'plate design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'material offset', ratio_measure> </pre>
plate_design_definition to any_surface (as moulded_surface)	#1: PATH		340	#1: extended_product_definition_shape <= product_definition_shape <-

Application Element	AIM element	Source	Rules	Reference Path
<p>#1: moulded surface is a surface</p> <p>#2: moulded surface is an external_instance_reference</p>	#2: PATH			<pre> shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'moulded surface'>] [shape_aspect = represented_definition <- PDR<'moulded surface representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', surface]]} #2: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'moulded surface'>] [EXT_INST_REF<shape_aspect, 'ship moulded forms', 'moulded form'>]]} </pre>
thickness	value_representation_item. value_component	43	294, 312	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'plate design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'thickness', positive_length_measure> </pre>
plate_design_definition to plate (as defined_for)	PATH		!only S[1:1] possible!	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'plate'>} </pre>
- from				

Application Element	AIM element	Source	Rules	Reference Path
STRUCTURAL_PART_DESIGN_DEFINITION				
mass	value_representation_item. value_component	43	294, 313	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'plate design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'mass', mass_measure>
plate_design_definition to shape_representation (as representations)	<i>see supertype</i>			
- from DEFINITION				
plate_design_definition to global_id (as id)	<i>see supertype</i>			
plate_design_definition to unit (as local_units)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PLATE_FUNCTIONAL_DEFINITION	extended_property_definition	218	342	{[CLASS<extended_property_definition, 'plate functional definition', 'structural part functional definition'>] [CLASS<extended_property_definition, 'structural part functional definition', 'functional definition'>] [CLASS<extended_property_definition, 'functional definition', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition, 'versionable object'>]]}
plate_functional_definition to plate (as defined_for)	PATH		!only S[1:1] possibl	extended_property_definition <= property_definition property_definition.definition -> characterized_definition

Application Element	AIM element	Source	Rules	Reference Path
			e!	= characterized_product_definition = product_definition {CLASS_ID<product_definition, 'plate'>}
the_function	descriptive_representation_item. description	45	341, 295	extended_property_definition <= property_definition = represented_definition <= PDR<'plate function parameters'> -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'function'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'standard plate') (descriptive_representation_item.description = 'doubling plate') (descriptive_representation_item.description = 'shedding plate') (descriptive_representation_item.description = 'clip') (descriptive_representation_item.description = 'watertight clip') (descriptive_representation_item.description = 'lug') (descriptive_representation_item.description = 'bracket') (descriptive_representation_item.description = 'face plate') (descriptive_representation_item.description = 'user defined')}
- from FUNCTIONAL_DEFINITION				
user_def_function	descriptive_representation_item. description	45	352	extended_property_definition <= property_definition = represented_definition <= PDR<'plate function parameters'> -> representation {[representation.name = '.UNUSED.']}

Application Element	AIM element	Source	Rules	Reference Path
				[representation.id = 'UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'user def function'} representation_item => descriptive_representation_item descriptive_representation_item.description
- from DEFINITION				
plate_functional_definition to global_id (as id)	<i>see supertype</i>			
plate_functional_definition to unit (as local_units)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PLATE_RELATIONSHIP	product_definition_relations hip	41		<i>see subtypes</i>
#1: plate_boundary_relationship				
plate_relationship to plate (as item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship. relating_product_definition -> product_definition {CLASS_ID<product_definition, 'plate'>}}
- from ITEM_RELATIONSHIP				
plate_relationship to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship. relating_product_definition -> product_definition [CLASS_ID<product_definition, 'plate'>] [EXT_INST_REF<product_definition, 'ship structures', 'plate'>]]}
plate_relationship to item (as item_2)	<i>see subtypes</i>			
plate_relationship to external_instance_reference (as external_item_2)	<i>see subtypes</i>			
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
plate_relationship to global_id (as id)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
plate_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROFILE	product_definition	41		{[CLASS<product_definition, 'profile', 'structural part'>] [CLASS<product_definition, 'structural part', 'part'> [CLASS<product_definition, 'part', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]]}
- from ITEM				
name	<i>see supertype</i>			
profile to external_reference (as documentation)	<i>see supertype</i>			
description	<i>see supertype</i>			
profile to ship (as ship_context)	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
profile to global_id (as id)	<i>see supertype</i>			
profile to definition (as definitions)	<i>see supertype</i>			
PROFILE_BOUNDARY_RELATIONSHIP	product_definition_relationship	41		{[CLASS<product_definition_relationship, 'profile boundary relationship', 'profile relationship'>] [CLASS<product_definition_relationship, 'profile relationship', 'structural part relationship'>] [CLASS<product_definition_relationship, 'structural part relationship', 'part relationship'>] [CLASS<product_definition_relationship, 'part relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]}

Application Element	AIM element	Source	Rules	Reference Path
profile_boundary_relationship to item (as item_2)	#1: product_definition_relations hip. related_product_definition	#1:41 #2:41 #3:41		#1: {product_definition_relationship. related_product_definition -> PDCD<'panel system'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {(CLASS_ID<shape_aspect, 'border'>) (CLASS_ID<shape_aspect, 'moulded surface'>)} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>}}
#1: related item is a panel system				
#2: related item is another profile	#2: product_definition_relations hip. related_product_definition			
#3: related item is a plate				
#4: related item is a seam	#3: product_definition_relations hip. related_product_definition			
#5: related item is an interior cutout	#4: PATH #5: PATH			#2: {product_definition_relationship {SAME_GROUP<product_definition_relationship. relating_product_definition, product_definition_relationship. related_product_definition, 'panel system'>} product_definition_relationship. related_product_definition -> PDCD<'profile'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'trace line'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>}} #3: product_definition_relationship {[SAME_GROUP<product_definition_relationship. relating_product_definition, product_definition_relationship. related_product_definition, 'panel system'>] [product_definition_relationship. related_product_definition -> PD_HAS_FUNCTION<'bracket'>]} product_definition_relationship. related_product_definition -> PDCD<'plate'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'moulded surface'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>}} #4: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'seam'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>} #5: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'interiour cutout'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>} </pre>
boundary_index	#1: shape_aspect	41		<pre> #1: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD<'panel system'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect </pre>

Application Element	AIM element	Source	Rules	Reference Path
				{(CLASS_ID<shape_aspect, 'border'>) (EXT_INST_REF<shape_aspect, 'ship structures', 'border'})}
displacement	value_representation_item. value_component	43	294, 314	product_definition_relationship product_definition_relationship.relating_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect <- {CLASS_ID<shape_aspect, 'border'} shape_aspect_relationship = {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>} represented_definition <- PDR<'profile boundary relationship design parameters'> -> Representation representation.items[i] -> VAL_REP_ITEM<'displacement', length_measure>
- from PROFILE_RELATIONSHIP				
profile_boundary_relationship to profile (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				
profile_boundary_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
profile_boundary_relationship to external_instance_reference (as external_item_2)	#1: product_definition_relations hip. related_product_definition	#1:41		#1: {product_definition_relationship. related_product_definition -> product_definition [CLASS_ID<product_definition, 'panel system'>] [EXT_INST_REF<product_definition, 'ship structures',
#1: related item is an external panel system				

Application Element	AIM element	Source	Rules	Reference Path
<p>#2: related item is an external profile (meaningless because it must be a profile of the same panel system)</p> <p>#3: related item is an external plate (meaningless because it must be a profile of the same panel system)</p> <p>#4: related item is an external seam</p> <p>#5: related item is an external interior cutout</p>	<p>#4: PATH</p> <p>#5: PATH</p>			<p>'panel system'>]]</p> <p>#4: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[CLASS_ID<shape_aspect, 'seam'>] [EXT_INST_REF<shape_aspect, 'ship structures', 'seam'>]]}</p> <p>#5: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[CLASS_ID<shape_aspect, 'interior cutout'>] [EXT_INST_REF<shape_aspect, 'ship structures', 'interior cutout'>]]}</p>
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
profile_boundary_relationship to global_id (as id)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
profile_boundary_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROFILE_CROSS_SECTION #1: explicit_profile_cross_section #2: library_profile_cross_section #3: parametric_profile_cross_section	shape_aspect	41		<i>see subtypes</i>
profile_cross_section to section_properties (as section_properties)	PATH		343	shape_aspect = represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = 'section properties'} property_definition_representation property_definition_representation.used_representation -> representation {CLASS_ID<representation, 'section properties'>}
designation	shape_aspect.name	41		
PROFILE_CURVE_TRACE_LINE	shape_aspect	41	344	{ROOT_CLASS<shape_aspect, 'profile curve trace line'>}
displacement	value_representation_item. value_component	43	294, 316	shape_aspect = represented_definition <- PDR<'profile curve trace line design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'displacement', length_measure>
profile_curve_trace_line to curve (as curve)	PATH		294, 315	shape_aspect = represented_definition <- PDR<'profile curve trace line design parameters'> -> representation

Application Element	AIM element	Source	Rules	Reference Path
				representation.items[i] -> GEO_REP_ITEM<'curve', curve>
PROFILE_DESIGN_DEFINITION	extended_product_definition_shape	218	346	{[CLASS<extended_product_definition_shape, 'plate design definition', 'structural part design definition'>] [CLASS<extended_product_definition_shape, 'structural part design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
profile_design_definiton to profile_boundary (as border) (OPT) #1: border is a line, point_on_curve or cartesian_point #2: border is a profile_boundary_relationship	#1: PATH #2: PATH		345	#1: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect = represented_definition <- PDR<'border representation'> -> representation representation.items[i] -> (GEO_REP_ITEM<'UNUSED.', line>) (GEO_REP_ITEM<'UNUSED.', point_on_curve>) (GEO_REP_ITEM<'UNUSED.', cartesian_point>)]} #2: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <- shape_aspect_relationship.relatng_shape_aspect

Application Element	AIM element	Source	Rules	Reference Path
				shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile boundary relationship'>}}
mirrored	descriptive_representation_item. description	45	317	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'profile design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'mirrored'> {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
profile_design_definition to profile_trace_line (as trace_line) #1: trace_line is a profile_curve_trace_line #2: trace_line is a profile_trace_line_relationship #3: trace_line is an external_instance_reference	#1: PATH #2: PATH #3: PATH		347	#1: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'trace line'>] [CLASS_ID<shape_aspect, 'profile curve trace line'>]} #2: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'trace line'>] [shape_aspect <= shape_aspect_relationship.relate_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile trace line relationship'>}}]} #3: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape

Application Element	AIM element	Source	Rules	Reference Path
				shape_aspect {[ROOT_CLASS<shape_aspect, 'trace line'>] [EXT_INST_REF<extended_product_definition_shape, 'ship moulded forms', 'moulded form'>]]}
profile_design_definition to twist_location (as twist)	PATH			extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {CLASS_ID<shape_aspect, 'twist location'>}
profile_design_definition to profile_cross_section (as cross_section)	PATH		348	extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {(shape_aspect.name = 'cross section') (CLASS_ID<shape_aspect, 'profile cross section'>)}
profile_design_definition to profile (as defined_for)	PATH		'only S[1:1] possible!	extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'profile'>}
- from STRUCTURAL_PART_DESIGN_DEFINITION				
mass	value_representation_item. value_component	43	294, 318	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'profile design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'mass', mass_measure>

Application Element	AIM element	Source	Rules	Reference Path
profile_design_definition to shape_representation (as representations) - from DEFINITION	<i>see supertype</i>			
profile_design_definition to global_id (as id)	<i>see supertype</i>			
profile_design_definition to unit (as local_units) - from VERSIONABLE_OBJECT	<i>see supertype</i>			
version_id	<i>see supertype</i>			
PROFILE_FUNCTIONAL_DEFINITION	extended_property_definition	218	350	{[CLASS<extended_property_definition, 'profile functional definition', 'structural part functional definition'>] [CLASS<extended_property_definition, 'structural part functional definition', 'functional definition'>] [CLASS<extended_property_definition, 'functional definition', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition, 'versionable object'>]]}
profile_functional_definition to profile (as defined_for)	PATH		!only S[1:1] possible!	extended_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'profile'>}
the_function	descriptive_representation_item. description	45	349, 295	extended_property_definition <= property_definition = represented_definition <= PDR<'profile function parameters'> -> representation {[representation.name = 'UNUSED.'] [representation.id = 'UNUSED.']} representation.items[i] -

Application Element	AIM element	Source	Rules	Reference Path
				> representation_item {representation_item.name = 'function'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'stiffener') (descriptive_representation_item.description = 'longitudinal stiffener') (descriptive_representation_item.description = 'transversal stiffener') (descriptive_representation_item.description = 'carling') (descriptive_representation_item.description = 'beam') (descriptive_representation_item.description = 'deck beam') (descriptive_representation_item.description = 'pillar') (descriptive_representation_item.description = 'hold pillar') (descriptive_representation_item.description = 'flange') (descriptive_representation_item.description = 'web') (descriptive_representation_item.description = 'frame') (descriptive_representation_item.description = 'user defined')}
- from FUNCTIONAL_DEFINITION				
user_def_function	descriptive_representation_i tem. description	45	353	extended_property_definition <= property_definition = represented_definition <- PDR<'profile function parameters'> -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] - > representation_item {representation_item.name = 'user def function'} representation_item => descriptive_representation_item descriptive_representation_item.description
- from DEFINITION				
profile_functional_definition to global_id (as id)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
profile_functional_definition to unit (as local_units)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROFILE_RELATIONSHIP	product_definition_relations hip	41		<i>see subtypes</i>
#1: profile_trace_line_relationship				
#2: profile_boundary_relationship				
profile_relationship to profile (as item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship. relating_product_definition -> product_definition {CLASS_ID<product_definition, 'profile'>}}
- from ITEM_RELATIONSHIP				
profile_relationship to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship.relying_product_definiti on -> product_definition [CLASS_ID<product_definition, 'profile'>] [EXT_INST_REF<product_definition, 'ship structures', 'profile'>]]}
profile_relationship to item (as item_2)	<i>see subtype</i>			
profile_relationship to external_instance_reference (as external_item_2)	<i>see subtype</i>			
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
profile_relationship to global_id (as id)	<i>see supertype</i>			
profile_relationship to definition (as definitions)(1)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROFILE_TRACE_LINE_RELATIONSHIP	product_definition_relations hip	41		{[CLASS<product_definition_relationship, 'profile trace line relationship', 'profile relationship'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<product_definition_relationship, 'profile relationship', 'structural part relationship'>] [CLASS<product_definition_relationship, 'structural part relationship', 'part relationship'>] [CLASS<product_definition_relationship, 'part relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]
profile_trace_line_relationship to item (as item_2) #1: related item is a panel system #2: related item is a seam #3: related item is an interior cutout	#1: product_definition_relationship. related_product_definition #2: PATH #3: PATH	#1:41		#1: {product_definition_relationship. related_product_definition -> PDCD<'panel system'> -> property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'border'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile trace line relationship'>}} #2: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition

Application Element	AIM element	Source	Rules	Reference Path
				<pre> property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[CLASS_ID<shape_aspect, 'seam'] [shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile trace line relationship'>}]}] #3: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'interiour cutout'} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'profile trace line relationship'>}</pre>
boundary_index	#1: shape_aspect	41		<pre> #1: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD<'panel system'> <- property_definition.definition property_definition => product_definition_shape</pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> product_definition_shape <- shape_aspect {(CLASS_ID<shape_aspect, 'border'>) (EXT_INST_REF<shape_aspect, 'ship structures', 'border')}</pre>
displacement	value_representation_item. value_component	43	294, 319	<pre> product_definition_relationship product_definition_relationship.relating_product_definitio n -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect <- {CLASS_ID<shape_aspect, 'trace line'} shape_aspect_relationship = represented_definition <- PDR<'profile trace line relationship design parameters' -> Representation representation.items[i] -> VAL_REP_ITEM<'displacement', length_measure></pre>
profile_trace_line_relationship to line (as auxiliary_line)	#1: PATH		294, 319	<pre> #1: product_definition_relationship product_definition_relationship.relating_product_definitio n -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect <- {CLASS_ID<shape_aspect, 'trace line'} shape_aspect_relationship = represented_definition <-</pre>

Application Element	AIM element	Source	Rules	Reference Path
				PDR<'profile trace line relationship design parameters' -> representation representation.items[i] -> GEO_REP_ITEM<'auxiliary line', line>
- from PROFILE_RELATIONSHIP				
profile_trace_line_relationship to profile (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				
profile_trace_line_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
profile_trace_line_relationship to external_instance_reference (as external_item_2)	#1: product_definition_relations hip. related_product_definition	#1:41		#1: {product_definition_relationship. related_product_definition -> product_definition [CLASS_ID<product_definition, 'panel system'>] [EXT_INST_REF<product_definition, 'ship structures', 'panel system'>]]}
#1: related item is an externally defined panel system	#2: PATH			#2: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'seam'>] [EXT_INST_REF<shape_aspect, 'ship structures', 'seam'>]]}
#2: related item is an externally defined seam	#3: PATH			
#3: related item is an externally defined interior cutout				

Application Element	AIM element	Source	Rules	Reference Path
				#3: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[CLASS_ID<shape_aspect, 'interiour cutout'] [EXT_INST_REF<shape_aspect, 'ship structures', 'interiour cutout']}]
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
profile_trace_line_relationship to global_id (as id)	<i>see supertype</i>			
profile_trace_line_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PROPRIETARY_PROFILE_CROSS_SECTION	shape_aspect	41		<i>see subtypes</i>
#1: flanged_plate_cross_section				
- from PROFILE_CROSS_SECTION				
proprietary_profile_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
ROUND_BAR_CROSS_SECTION	shape_aspect	41	351	{[CLASS<shape_aspect, 'round bar cross section', 'bar profile cross section'] [CLASS<shape_aspect, 'bar profile cross section', 'parametric profile cross section'] [CLASS<shape_aspect, 'parametric profile cross section',

Application Element	AIM element	Source	Rules	Reference Path
				'profile cross section'] [ROOT_CLASS<shape_aspect, 'profile cross section']]
diameter	value_representation_item. value_component	43	294, 320	shape_aspect = represented_definition <- PDR<'round bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'diameter', positive_length_measure>
- from PROFILE_CROSS_SECTION				
round_bar_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
SECTION_PROPERTIES	representation	43		{ROOT_CLASS< representation, 'section properties'>}
nominal_mass_per_len	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'nominal mass per length', mass_measure>
area	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'area', area_measure>
na_u	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'na u', positive_length_measure>
na_v	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'na v', positive_length_measure>
moi_u	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'moi u', context_dependent_measure>
moi_v	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'moi v', context_dependent_measure>
moi_uv	value_representation_item.	43	294,	representation

Application Element	AIM element	Source	Rules	Reference Path
	value_component		321	representation.items[i] -> VAL_REP_ITEM<'moi uv', context_dependent_measure>
tr	value_representation_item. value_component	43	294, 321	representation representation.items[i] -> VAL_REP_ITEM<'tr', context_dependent_measure>
SQUARE_TUBE_CROSS_SECTION	shape_aspect	41		{[CLASS<shape_aspect, 'square tube cross section', 'non circular profile cross section'] [CLASS<shape_aspect, 'non circular profile cross section', 'parametric profile cross section'] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'] [ROOT_CLASS<shape_aspect, 'profile cross section']}]
depth	value_representation_item. value_component	43	294, 322	shape_aspect = represented_definition <- PDR<'square tube cross section design parameters' -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 322	shape_aspect = represented_definition <- PDR<'square tube cross section design parameters' -> representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
wall_thk	value_representation_item. value_component	43	294, 322	shape_aspect = represented_definition <- PDR<'square tube cross section design parameters' -> representation representation.items[i] -> VAL_REP_ITEM<'wall thk', positive_length_measure>
- from PROFILE_CROSS_SECTION				
square_tube_cross_section to section_properties (as section_properties)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
designation	<i>see supertype</i>			
STRUCTURAL_PART	product_definition	41		<i>see subtypes</i>
#1: plate				
#2: profile				
#3: corrugated_structure				
- from ITEM				
name	product_definition.name	41		
structural_part to external_reference (as documentation)	PATH			<pre> product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference { applied_document_reference <= document_reference document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = 'UNUSED. ']} </pre>
description	product_definition.description	41		
structural_part to ship (as ship_context)	PATH			<pre> product_definition product_definition.formation -> product_definition_formation { product_definition_formation.id = 'ship structure' } product_definition_formation.of_product -> product { CLASS_ID<product, 'ship'> } </pre>
- from DEFINABLE_OBJECT				
structural_part to global_id (as id)	product_definition.id	41	293, 296	
structural_part to definition (as definitions)	PATH			product_definition =

Application Element	AIM element	Source	Rules	Reference Path
				characterized_product_definition = characterized_definition <- property_definition.definition property_definition
STRUCTURAL_PART_DESIGN_DEFINITION #1: plate_design_definition #2: profile_design_definition #3: corrugated_part_design_definition	extended_product_definition_shape	218		<i>see subtypes</i>
mass	value_representation_item. value_component	43	294	<i>see subtypes</i>
structural_part_design_definition to shape_representation (as representations)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition <- represented_definition <- PDR<'representation' property_definition_representation.used_representation -> representation => shape_representation
structural_part_design_definition to structural_part (as defined_for)	PATH			<i>see subtypes</i>
- from DEFINITION				
structural_part_design_definition to global_id (as id)	extended_product_definition_shape. id	218	293, 300	
structural_part_design_definition to unit (as local_units)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition {property_definition.name = 'UNUSED.'} property_definition <-

Application Element	AIM element	Source	Rules	Reference Path
				<pre> represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = '.UNUSED.']} </pre>
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	358	<pre> extended_product_definition_shape <- applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id </pre>
STRUCTURAL_PART_FUNCTIONAL_DEFINITION (ABS)	extended_property_definition	218		<i>see subtypes</i>
#1: plate_functional_definition				
#2: profile_functional_definition				
#3: corrugated_part_functional_definition				
structural_part_functional_definition to	PATH			<i>see subtypes</i>

Application Element	AIM element	Source	Rules	Reference Path
structural_part (as defined_for)				
the_function	descriptive_representation_item. description	45		<i>see subtypes</i>
- from FUNCTIONAL_DEFINITION				
user_def_function	descriptive_representation_item. description	45		<i>see subtypes</i>
- from DEFINITION				
structural_part_functional_definition to global_id (as id)	extended_property_definition.id	218	293, 298	
structural_part_functional_definition to unit (as local_units)				
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	358	extended_property_definition = identification_item <- applied_identification_assignment.items[I] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
STRUCTURAL_PART_RELATIONSHIP				
#1: structural_part_symmetry_relationship				
#2: plate_relationship				
#3: profile_relationship				
structural_part_relationship to structural_part (as item_1)	product_definition_relationship. relating_product_definition	41		<i>see subtypes</i>
- from ITEM_RELATIONSHIP				

Application Element	AIM element	Source	Rules	Reference Path
structural_part_relationship to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition			<i>see subtypes</i>
structural_part_relationship to item (as item_2)	<i>see subtypes</i>			
structural_part_relationship to external_instance_reference (as external_item_2)	<i>see subtypes</i>			
contxt	product_definition_relations hip. description	41		
- from DEFINABLE_OBJECT				
structural_part_relationship to global_id (as id)	product_definition_relations hip.id	41	293, 299	
structural_part_relationship to definition (as definitions)	PATH			product_definition_relationship = characterized_product_definition = characterized_definition <- property_definition.definition property_definition
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	358	product_definition_relationship = identification_item <- applied_identification_assignment.items[I] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
STRUCTURAL_PART_SYMMETRY_RELATIONSHIP	product_definition_relations hip	41	297	{[CLASS<product_definition_relationship, 'structural part symmetry relationship', 'structural part relationship'>] [CLASS<product_definition_relationship, 'structural part relationship', 'part relationship'>] [CLASS<product_definition_relationship,

Application Element	AIM element	Source	Rules	Reference Path
				'part relationship', 'item relationship'] [CLASS<product_definition_relationship, 'item relationship', 'definable object'] [ROOT_CLASS<product_definition_relationship, 'definable object'] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'] [ROOT_CLASS<product_definition_relationship, 'versionable object']}]
structural_part_symmetry_relationship to plane (as mirroring_plane)	PATH		294, 354	product_definition_relationship = {product_definition_relationship.name = '.UNUSED.'} characterized_product_definition = characterized_definition <- property_definition.definition property_definition = {property_definition.name = '.UNUSED.'} represented_definition <- PDR<'structural part symmetry relationship design paramet ers'> -> representation representation.items[i] -> GEO_REP_ITEM<'mirroring plane', plane>
structural_part_symmetry_relationship to structural_part (as item_2)	product_definition_relations hip. related_product_definition	41		{product_definition_relationship.related_product_definitio n -> product_definition (CLASS_ID<product_definition, 'plate'>) (CLASS_ID<product_definition, 'profile'>) (CLASS_ID<product_definition, 'corrugated part'>)}]
- from STRUCTURAL_PART_RELATIONSHIP				
structural_part_symmetry_relationship to structural_part (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				

Application Element	AIM element	Source	Rules	Reference Path
structural_part_symmetry_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
structural_part_symmetry_relationship to external_instance_reference (as external_item_2)	product_definition_relationship. related_product_definition	41		{product_definition_relationship.related_product_definition -> product_definition ([CLASS_ID<product_definition, 'plate'>] [EXT_INST_REF<product_definition, 'ship structures', 'plate'>]) ([CLASS_ID<product_definition, 'profile'>] [EXT_INST_REF<product_definition, 'ship structures', 'profile'>]) ([CLASS_ID<product_definition, 'corrugated part'>] [EXT_INST_REF<product_definition, 'ship structures', 'corrugated part'>])}]}
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
structural_part_symmetry_relationship to global_id (as id)	<i>see supertype</i>			
structural_part_symmetry_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
TWIST_LOCATION	shape_aspect	41	355	{ROOT_CLASS<shape_aspect, 'twist location'>}
twist_location to point_on_curve (as location)	PATH		294, 323	shape_aspect = represented_definition <- PDR<'twist location design parameters'> -> representation => representation.items[i] -> GEO_REP_ITEM<'location', point_on_curve>
twist_location to direction (as direction)	PATH		294, 323	shape_aspect = represented_definition <- PDR<'twist location design parameters'> ->

Application Element	AIM element	Source	Rules	Reference Path
				representation => representation.items[i] -> GEO_REP_ITEM<'direction', direction>
T_BAR_CROSS_SECTION	shape_aspect	41	356	{[CLASS<shape_aspect, 't bar cross section', 'flanged profile cross section'>] [CLASS<shape_aspect, 'flanged profile cross section', 'parametric profile cross section'>] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]}
depth	value_representation_item. value_component	43	294, 324	shape_aspect = represented_definition <- PDR<'t bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 324	shape_aspect = represented_definition <- PDR<'t bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
web_thk	value_representation_item. value_component	43	294, 324	shape_aspect = represented_definition <- PDR<'t bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'web thk', positive_length_measure>
flange_thk	value_representation_item. value_component	43	294, 324	shape_aspect = represented_definition <- PDR<'t bar cross section design parameters'> -> representation representation.items[i] ->

Application Element	AIM element	Source	Rules	Reference Path
				VAL_REP_ITEM<'flange thk', positive_length_measure>
radius	value_representation_item. value_component	43	294, 324	shape_aspect = represented_definition <- PDR<'t bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius', positive_length_measure>
k	value_representation_item. value_component	43	294, 324	shape_aspect = represented_definition <- PDR<'t bar cross section design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'k', positive_length_measure>
- from PROFILE_CROSS_SECTION				
t_bar_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			
W_SHAPE_CROSS_SECTION	shape_aspect	41	357	{[CLASS<shape_aspect, 'w shape cross section', 'flanged profile cross section'>] [CLASS<shape_aspect, 'flanged profile cross section', 'parametric profile cross section'>] [CLASS<shape_aspect, 'parametric profile cross section', 'profile cross section'>] [ROOT_CLASS<shape_aspect, 'profile cross section'>]]}
depth	value_representation_item. value_component	43	294, 325	shape_aspect = represented_definition <- PDR<'w shape cross section'> -> representation representation.items[i] -> VAL_REP_ITEM<'depth', positive_length_measure>
width	value_representation_item. value_component	43	294, 325	shape_aspect = represented_definition <- PDR<'w shape cross section'> ->

Application Element	AIM element	Source	Rules	Reference Path
				representation representation.items[i] -> VAL_REP_ITEM<'width', positive_length_measure>
web_thk	value_representation_item. value_component	43	294, 325	shape_aspect = represented_definition <- PDR<'w shape cross section'> -> representation representation.items[i] -> VAL_REP_ITEM<'web thk', positive_length_measure>
flange_thk	value_representation_item. value_component	43	294, 325	shape_aspect = represented_definition <- PDR<'w shape cross section'> -> representation representation.items[i] -> VAL_REP_ITEM<'flange thk', positive_length_measure>
radius	value_representation_item. value_component	43	294, 325	shape_aspect = represented_definition <- PDR<'w shape cross section'> -> representation representation.items[i] -> VAL_REP_ITEM<'radius', positive_length_measure>
k	value_representation_item. value_component	43	294, 325	shape_aspect = represented_definition <- PDR<'w shape cross section'> -> representation representation.items[i] -> VAL_REP_ITEM<'k', positive_length_measure>
- from PROFILE_CROSS_SECTION				
w_shape_cross_section to section_properties (as section_properties)	<i>see supertype</i>			
designation	<i>see supertype</i>			

Table 16 - Mapping table for structural_systems UoF

Application Element	AIM element	Source	Rules	Reference Path
BUILT_PROFILE	[product_definition] [group]	41		<pre> {[LINK_TO_GROUP<product_definition>] [CLASS<product_definition, 'built profile', 'profile'>] [CLASS<product_definition, 'profile', 'structural part'>] [CLASS<product_definition, 'structural part', 'part'>] [CLASS<product_definition, 'part', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>] [CLASS<product_definition, 'built profile', 'structural system'>] [CLASS<product_definition, 'structural system', 'system'>] [CLASS<product_definition, 'system', 'item structure'>] [CLASS<product_definition, 'item structure', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>] [CLASS<product_definition, 'item structure', 'versionable object'>] [ROOT_CLASS<product_definition, 'versionable object'>]} </pre>
the_class	group.name	41		<pre> product_definition classification_item = product_definition classification_item <- applied_classification_assignment.items[i] applied_classification_assignment applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'class membership'} classification_assignment.assigned_classification -> group </pre>

Application Element	AIM element	Source	Rules	Reference Path
				group.name { group.name = 'built profile' }
- from STRUCTURAL_SYSTEM				
user_def_class	<i>see supertype</i>			
- from ITEM				
description	<i>see supertype</i>			
name	<i>see supertype</i>			
built_profile to external_reference (as documentation)	<i>see supertype</i>			
built_profile to ship (as ship_context)	<i>see supertype</i>			
- from ITEM_STRUCTURE				
built_profile to external_instance_reference (as external_items)	PATH			group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'plate'>] [EXT_INST_REF<product_definition, 'ship structures', 'plate'>]) ([CLASS_ID<product_definition, 'profile'>] [EXT_INST_REF<product_definition, 'ship structures', 'profile'>])}]}
built_profile to external_instance_reference (as external_relationships)	PATH			group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition_relationship, 'structural part relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural part relationship'>])}]}
built_profile to item (as items)	PATH			group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'plate'>] ([CLASS_ID<product_definition, 'profile'>])}]}
built_profile to item_relationship (as relationships)	PATH			group <- GROUPS<product_definition_relationship, 'item structure'>

Application Element	AIM element	Source	Rules	Reference Path
				{ CLASS_ID<product_definition, 'structural part relationship'> }
- from DEFINABLE_OBJECT				
built_profile to global_id (as id)	<i>see supertype</i>			
built_profile to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORRUGATED_STRUCTURE	[product_definition] [group]	41		{ [LINK_TO_GROUP<product_definition> [CLASS<product_definition, 'corrugated structure', 'structural system'>] [CLASS<product_definition, 'structural system', 'system'>] [CLASS<product_definition, 'system', 'item structure'>] [CLASS<product_definition, 'item structure', 'definable object'>] [CLASS<product_definition, 'item structure', 'versionable object'>] [ROOT_CLASS<product_definition, 'versionable object'>] [CLASS<product_definition, 'system', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]] }
the_class	group.name	41		product_definition classification_item = product_definition classification_item <- applied_classification_assignment.items[i] applied_classification_assignment applied_classification_assignment <= classification_assignment { classification_assignment.role -> classification_role classification_role.name = 'class membership' } classification_assignment.assigned_classification -> group group.name

Application Element	AIM element	Source	Rules	Reference Path
				{group.name = 'corrugated structure'}
- from STRUCTURAL_SYSTEM				
user_def_class	<i>see supertype</i>			
- from ITEM				
description	<i>see supertype</i>			
name	<i>see supertype</i>			
corrugated_structure to external_reference (as documentation)	<i>see supertype</i>			
corrugated_structure to ship (as ship_context)	<i>see supertype</i>			
- from ITEM_STRUCTURE				
corrugated_structure to external_instance_reference (as external_items)	PATH			group <- GROUPS<product_definition, 'item structure'> {[CLASS_ID<product_definition, 'plate'>] [EXT_INST_REF<product_definition, 'ship structures', 'plate'>]] ([CLASS_ID<product_definition, 'corrugated part'>] [EXT_INST_REF<product_definition, 'ship structures', 'corrugated part'>])}]
corrugated_structure to external_instance_reference (as external_relationships)	PATH			group <- GROUPS<product_definition_relationship, 'item structure'> {[CLASS_ID<product_definition_relationship, 'plate relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural part relationship'>]]}
corrugated_structure to item (as items)	PATH			group <- GROUPS<product_definition, 'item structure'> {[CLASS_ID<product_definition, 'plate'>] (CLASS_ID<product_definition, 'corrugated part'>)]}
corrugated_structure to item_relationship (as relationships)	PATH			group <- GROUPS<product_definition_relationship, 'item structure'> {CLASS_ID<product_definition_relationship,

Application Element	AIM element	Source	Rules	Reference Path
				'structural part relationship'>}
- from DEFINABLE_OBJECT				
corrugated_structure to global_id (as id)	<i>see supertype</i>			
corrugated_structure to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORRUGATED_STRUCTURE_BOUNDARY_RELATIONSHIP	product_definition_relationship	41		{[CLASS<product_definition_relationship, 'corrugated structure boundary relationship', 'corrugated structure relationship'>] [CLASS<product_definition_relationship, 'corrugated structure relationship', 'structural system relationship'>] [CLASS<product_definition_relationship, 'structural system relationship', 'system relationship'>] [CLASS<product_definition_relationship, 'system relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]}
corrugated_structure_boundary_relationship to item (as item_2) #1: related item is a corrugated_structure #2: related item is a panel_system #3: related item is a seam	#1: product_definition_relationship. related_product_definition #2: product_definition_relationship.	#1: 41 #2: 41		#1: {product_definition_relationship. related_product_definition -> PDCD<'corrugated structure'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape

Application Element	AIM element	Source	Rules	Reference Path
	related_product_definition #3: PATH			<pre> shape_aspect <- {CLASS_ID<shape_aspect, 'moulded surface'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'corrugated structure boundary relationship'>}} #2: {[product_definition_relationship. related_product_definition -> PDCD<'panel system'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'moulded surface'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'corrugated structure boundary relationship'>}}}] #3: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'seam'>}</pre>

Application Element	AIM element	Source	Rules	Reference Path
				shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'corrugated structure boundary relationship'>}
displacement	value_representation_item. value_component	43	365, 367	product_definition_relationship product_definition_relationship.relating_product_definitio n -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect <- {CLASS_ID<shape_aspect, 'border'} shape_aspect_relationship = {CLASS_ID<shape_aspect_relationship, 'panel system boundary relationship'>} represented_definition <- PDR<'corrugated structure boundary relationship design pa rameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'displacement', length_measure>
- from CORRUGATED_STRUCTURE_RELATIONS HIP				
corrugated_structure_boundary_relationship to corrugated_structure (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				
corrugated_structure_boundary_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
corrugated_structure_boundary_relationship to	#1:	#1: 41		#1: {product_definition_relationship.

Application Element	AIM element	Source	Rules	Reference Path
external_instance_reference (as external_item_2) #1: related item is a corrugated_structure #2: related item is a panel_system #3: related item is a seam	product_definition_relations hip. related_product_definition #2: product_definition_relations hip. related_product_definition #3: PATH	#2: 41		related_product_definition -> product_definition [CLASS_ID<product_definition, 'corrugated structure'>] [EXT_INST_REF<product_definition, 'ship structures', 'corrugated structure'>]] #2: {product_definition_relationship. related_product_definition -> product_definition [CLASS_ID<product_definition, 'panel system'>] [EXT_INST_REF<product_definition, 'ship structures', 'panel system'>]] #3: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'seam'>] [EXT_INST_REF<shape_aspect, 'ship structures', 'seam'>]]}
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
corrugated_structure_boundary_relationship to global_id (as id)	<i>see supertype</i>			
corrugated_structure_boundary_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
version_id	<i>see supertype</i>			
CORRUGATED_STRUCTURE_DESIGN_DEFINITION	extended_product_definition_shape	218	369	{[CLASS<extended_product_definition_shape, 'corrugated structure design definition', 'structural system design definition'>] [CLASS<extended_product_definition_shape, 'structural system design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
corrugated_structure_design_definition to corrugated_structure_boundary (as border) #1: border is a structural_system_adjacency_relationship #2: border is a corrugated_structure_boundary_relationship	#1: PATH #2: PATH		368	#1: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <= shape_aspect_relationship.relate_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'structural system adjacency relationship'>}]}} #2: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <= shape_aspect_relationship.relate_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'corrugated structure boundary relationship'>}]}}}

Application Element	AIM element	Source	Rules	Reference Path
corrugation_width	value_representation_item. value_component	43	365, 372	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'corrugated structure design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'corrugation width', length_measure>
thickness	value_representation_item. value_component	43	365, 371	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'corrugated structure design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'thickness', length_measure>
corrugated_structure_design_definition to structural_system_moulded_surface (as moulded_surface) #1: moulded_surface is a surface #2: moulded_surface is an external_instance_reference	#1: PATH #2: PATH		370	#1: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'moulded surface'>] [shape_aspect = represented_definition <- PDR<'moulded surface representation'> -> representation representation.items[i] -> GEO_REP_ITEM<'UNUSED.', surface]]} #2: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'moulded surface'>] [EXT_INST_REF<shape_aspect, 'ship moulded forms', 'moulded form'>]]}
corrugated_structure_design_definition to	PATH		!only	extended_product_definition_shape <=

Application Element	AIM element	Source	Rules	Reference Path
corrugated_structure (as defined_for)			S[1:1] possible!	product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'corrugated structure'>}
- from STRUCTURAL_SYSTEM_DESIGN_DEFINIT ION				
tightness	descriptive_representation_i tem. description	41	371, 363	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'corrugated structure design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'tightness'> {(descriptive_representation_item.description = 'unspecified') (descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'gas tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'user defined')}
user_defined_tightness	descriptive_representation_i	41	372	extended_product_definition_shape <=

Application Element	AIM element	Source	Rules	Reference Path
	tem. description			product_definition_shape <= property_definition = represented_definition <- PDR<'corrugated structure design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'user defined tightness'>
corrugated_structure_design_definition to shape_representation (as representations)	<i>see supertype</i>			
- from DEFINITION				
corrugated_structure_design_definition to unit (as local_units)	<i>see supertype</i>			
corrugated_structure_design_definition to global_id (as id)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORRUGATED_STRUCTURE_FUNCTIONAL DEFINITION	extended_property_definition	218	373	{[CLASS<extended_property_definition, 'corrugated structure functional definition', 'structural system functional definition'>] [CLASS<extended_property_definition, 'structural system functional definition', 'functional definition'>] [CLASS<extended_property_definition, 'functional definition', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition, 'versionable object'>]]}
the_function	descriptive_representation_item. description	45	362, 375	extended_property_definition <= property_definition = represented_definition <- PDR<'corrugated structure function parameters'> -> representation

Application Element	AIM element	Source	Rules	Reference Path
				<pre> representation.items[i] -> representation_item {representation_item.name = 'function'} representation_item => descriptive_representation_item {[(descriptive_representation_item.description = 'deck') (descriptive_representation_item.description = 'strength deck') (descriptive_representation_item.description = 'accomodation deck') (descriptive_representation_item.description = 'main deck') (descriptive_representation_item.description = 'weather deck') (descriptive_representation_item.description = 'navigation deck') (descriptive_representation_item.description = 'platform deck') (descriptive_representation_item.description = 'deck in superstructure') (descriptive_representation_item.description = 'bulkhead') (descriptive_representation_item.description = 'longitudinal bulkhead') (descriptive_representation_item.description = 'transversal bulkhead') (descriptive_representation_item.description = 'strength bulkhead') (descriptive_representation_item.description = 'wash bulkhead') (descriptive_representation_item.description = 'collision bulkhead') (descriptive_representation_item.description = 'hold bulkhead') (descriptive_representation_item.description = 'tank bulkhead') </pre>

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'wing bulkhead') (descriptive_representation_item.description = 'superstructure front bulkhead') (descriptive_representation_item.description = 'superstructure side bulkhead') (descriptive_representation_item.description = 'superstructure aft bulkhead') (descriptive_representation_item.description = 'stool') (descriptive_representation_item.description = 'wall') (descriptive_representation_item.description = 'hatch cover') (descriptive_representation_item.description = 'user defined'))}]
corrugated_structure_functional_definition to corrugated_structure (as defined_for)	PATH			extended_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'corrugated structure'>}
- from FUNCTIONAL_DEFINITION				
user_def_function	descriptive_representation_i tem. description	45	374	extended_property_definition <= property_definition = represented_definition <= PDR<'corrugated structure function parameters'> -> representation {[representation.name = '.UNUSED.'] [representation.id = '.UNUSED.']} representation.items[i] -> DSC_REP_ITEM<'user def function'>
- from DEFINITION				
corrugated_structure_functional_definition to	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
unit (as local_units)				
corrugated_structure_functional_definition to global_id (as id)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
CORRUGATED_STRUCTURE_RELATIONS HIP	product_definition_relations hip	41		<i>see subtypes</i>
#1: corrugated_structure_boundary_relationship				
corrugated_structure_relationship to corrugated_structure (as item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship.relying_product_definit on -> product_definition { CLASS_ID<product_definition, 'corrugated structure'> } }
- from ITEM_RELATIONSHIP				
corrugated_structure_relationship to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship.relying_product_definit on -> product_definition [CLASS<product_definition, 'corrugated structure'>] [EXT_INST_REF<product_definition, 'ship structures', 'corrugated structure'>] }
corrugated_structure_relationship to item (as item_2)	<i>see subtype</i>			
corrugated_structure_relationship to external_instance_reference (as external_item_2)	<i>see subtype</i>			
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
corrugated_structure_relationship to global_id (as id)	<i>see supertype</i>			
corrugated_structure_relationship to definition (as definitions)	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PANEL_SYSTEM	[product_definition] [group]	41		{[LINK_TO_GROUP<product_definition>] [CLASS<product_definition, 'panel system', 'structural system'>] [CLASS<product_definition, 'structural system', 'system'>] [CLASS<product_definition, 'system', 'item structure'>] [CLASS<product_definition, 'item structure', 'definable object'>] [CLASS<product_definition, 'item structure', 'versionable object'>] [ROOT_CLASS<product_definition, 'versionable object'>] [CLASS<product_definition, 'system', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]}
the_class	group.name	41		product_definition classification_item = product_definition classification_item <- applied_classification_assignment.items[i] applied_classification_assignment applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'class membership'} classification_assignment.assigned_classification -> group group.name {group.name = 'panel system'}
- from STRUCTURAL_SYSTEM				
user_def_class	<i>see supertype</i>			
- from ITEM				
description	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
name	<i>see supertype</i>			
panel_system to external_reference (as documentation)	<i>see supertype</i>			
panel_system to ship (as ship_context) - from ITEM_STRUCTURE	<i>see supertype</i>			
panel_system to external_instance_reference (as external_items)	PATH		398	<pre> group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'structural part'>] [EXT_INST_REF<product_definition, 'ship structures', 'structural part'>]) ([CLASS_ID<product_definition, 'panel system'>] [EXT_INST_REF<product_definition, 'ship structures', 'panel system'>])}}</pre>
panel_system to external_instance_reference (as external_relationships)	PATH			<pre> group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition_relationship, 'structural part relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural part relationship'>]) ([CLASS_ID<product_definition_relationship, 'structural system relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural system relationship'>])}}</pre>
panel_system to item (as items)	PATH		398	<pre> group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'structural part'>] ([CLASS_ID<product_definition, 'panel system'>])}</pre>
panel_system to item_relationship (as relationships)	PATH			<pre> group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition, 'structural part relationship'>])}</pre>

Application Element	AIM element	Source	Rules	Reference Path
				(CLASS_ID<product_definition, 'structural system relationship'>))
- from DEFINABLE_OBJECT				
panel_system to global_id (as id)	<i>see supertype</i>			
panel_system to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PANEL_SYSTEM_BOUNDARY_RELATIONSHIP	product_definition_relationship	41		{[CLASS<product_definition_relationship, 'panel system boundary relationship', 'panel system relationship'>] [CLASS<product_definition_relationship, 'panel system relationship', 'structural system relationship'>] [CLASS<product_definition_relationship, 'structural system relationship', 'system relationship'>] [CLASS<product_definition_relationship, 'system relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]}
panel_system_boundary_relationship to item (as item_2) #1: related item is a panel_system #2: related item is a profile #3: related item is a seam	#1: product_definition_relationship. related_product_definition #2: product_definition_relationship.	#1: 41 #2: 41		#1: {product_definition_relationship. related_product_definition -> PDCD<'panel system'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape

Application Element	AIM element	Source	Rules	Reference Path
	related_product_definition #3: PATH			<pre> shape_aspect <- {CLASS_ID<shape_aspect, 'moulded surface'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'panel system boundary relationship'>}} #2: {[product_definition_relationship. related_product_definition -> PDCD<'profile'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {(CLASS_ID<shape_aspect, 'top line'>) (CLASS_ID<shape_aspect, 'profile cross section'>)} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'panel system boundary relationship'>}}}] #3: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- </pre>

Application Element	AIM element	Source	Rules	Reference Path
				{CLASS_ID<shape_aspect, 'seam'>} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'panel system boundary relationship'>}
displacement	value_representation_item. value_component	43	365, 376	product_definition_relationship product_definition_relationship.relating_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect <- {CLASS_ID<shape_aspect, 'border'>} shape_aspect_relationship = {CLASS_ID<shape_aspect_relationship, 'panel system boundary relationship'>} represented_definition <- PDR<'panel system boundary relationship design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'displacement', length_measure>
- from PANEL_SYSTEM_RELATIONSHIP				
panel_system_boundary_relationship to panel_system (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				
panel_system_boundary_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
panel_system_boundary_relationship to external_instance_reference	#1: product_definition_relations	#1: 41		#1: {product_definition_relationship. related_product_definition ->

Application Element	AIM element	Source	Rules	Reference Path
(as external_item_2) #1: related item is a panel_system #2: related item is a profile #3: related item is a seam	hip. related_product_definition #2: product_definition_relations hip. related_product_definition #3: PATH	#2: 41		product_definition [CLASS_ID<product_definition, 'panel system'>] [EXT_INST_REF<product_definition, 'ship structures', 'panel system'>]] #2: {product_definition_relationship. related_product_definition -> product_definition [CLASS_ID<product_definition, 'profile'>] [EXT_INST_REF<product_definition, 'ship structures', 'profile'>]] #3: product_definition_relationship product_definition_relationship.related_product_definition -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'seam'>] [EXT_INST_REF<shape_aspect, 'ship structures', 'seam'>]]}
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
panel_system_boundary_relationship to global_id (as id)	<i>see supertype</i>			
panel_system_boundary_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
PANEL_SYSTEM_CURVE_BOUNDARY	shape_aspect	41	377	{[ROOT_CLASS<shape_aspect, 'panel system curve boundary'>]}
displacement	value_representation_item. value_component	43	365, 378	shape_aspect = represented_definition <- PDR<'panel system curve boundary design parameters'> -> representation representation.items[i] -> VAL_REP_ITEM<'displacement', length_measure>
panel_system_curve_boundary to bounded_curve (as curve)	PATH		365, 379	shape_aspect = represented_definition <- PDR<'panel system curve boundary design parameters'> -> representation representation.items[i] -> GEO_REPRESENTATION_ITEM<'curve', bounded_curve>
PANEL_SYSTEM_DESIGN_DEFINITION	extended_product_definition _shape	218	383	{[CLASS<extended_product_definition_shape, 'panel system design definition', 'structural system design definition'>] [CLASS<extended_product_definition_shape, 'structural system design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]}
material_offset	value_representation_item. value_component	43	365, 380	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'panel system design parameters'> -> representation representation.items[i] -> VAL_REPRESENTATION_ITEM<'material offset', ratio_measure>
stiffener_spacing	value_representation_item.	43	365,	extended_product_definition_shape <=

Application Element	AIM element	Source	Rules	Reference Path
	value_component		381	product_definition_shape <= property_definition = represented_definition <- PDR<'panel system design parameters'> -> representation representation.items[i] -> VAL_REPRESENTATION_ITEM<'stiffener spacing', length_measure>
thickness	value_representation_item. value_component	43	365, 380	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'panel system design parameters'> -> representation representation.items[i] -> VAL_REPRESENTATION_ITEM<'thickness', length_measure>
panel_system_design_definition to panel_system (as defined_for)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'panel system'>}
panel_system_design_definition to structural_system_moulded_surface (as moulded_surface) #1: moulded_surface is a surface #2: moulded_surface is an external_instance_reference	#1: PATH #2: PATH		384	#1: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'moulded surface'>] [shape_aspect = represented_definition <- PDR<'moulded surface representation'> -> representation representation.items[i] -> GEO_REPRESENTATION_ITEM<'.UNUSED.', surface]]}

Application Element	AIM element	Source	Rules	Reference Path
				#2: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'moulded form'>] [EXT_INST_REF<shape_aspect, 'ship moulded forms', 'moulded form'>]}
panel_system_design_definition to panel_system_boundary (as border) #1: panel_system_boundary is a panel_system_curve_boundary #2: panel_system_boundary is a panel_system_plane_boundary #3: panel_system_boundary is a structural_system_adjacency_relationship #4: panel_system_boundary is a panel_system_boundary_relationship	#1: PATH #2: PATH #3: PATH #4: PATH		382	#1: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'panel system curve boundary'>] [ROOT_CLASS<shape_aspect, 'border'>]} #2: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[CLASS_ID<shape_aspect, 'panel system plane boundary'>] [ROOT_CLASS<shape_aspect, 'border'>]} #3: extended_product_definition_shape <= product_definition_shape <= shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <= shape_aspect_relationship.relatng_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship,

Application Element	AIM element	Source	Rules	Reference Path
				<pre>'structural system adjacency relationship'>}}}</pre> <pre>#4: extended_product_definition_shape <= product_definition_shape <- shape_aspect.of_shape shape_aspect {[ROOT_CLASS<shape_aspect, 'border'>] [shape_aspect <- shape_aspect_relationship.relatng_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'panel system boundary relationship'>}]}}</pre>
- from STRUCTURAL_SYSTEM_DESIGN_DEFINIT ION				
tightness	descriptive_representation_i tem. description	41	363, 380	<pre>extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'panel system design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'tightness'> {((descriptive_representation_item.description = 'unspecified') (descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'gas tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'water tight'))</pre>

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'user defined'))}
user_defined_tightness	descriptive_representation_item. description	41	381	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'panel system design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'user defined tightness'>
panel_system_design_definition to shape_representation (as representations)	<i>see supertype</i>			
- from DEFINITION				
panel_system_design_definition to unit (as local_units)	<i>see supertype</i>			
structural_system_design_definition to global_id (as id)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PANEL_SYSTEM_FUNCTIONAL_DEFINITION	extended_property_definition	218	385	{[CLASS<extended_property_definition,, 'panel system functional definition', 'structural system functional definition'>] [CLASS<extended_property_definition,, 'structural system functional definition', 'functional definition'>] [CLASS<extended_property_definition,, 'functional definition', 'definition'>] [CLASS<extended_property_definition,, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition,, 'versionable object'>]]}

Application Element	AIM element	Source	Rules	Reference Path
the_function	descriptive_representation_item. description	45	362, 386	<pre> extended_property_definition <= property_definition = represented_definition <- PDR -> representation {[representation.name = '.UNUSED.']} [representation.id = '.UNUSED.']} representation.items[i] -> representation_item {representation_item.name = 'function'} representation_item => descriptive_representation_item {[descriptive_representation_item.description = 'deck'] (descriptive_representation_item.description = 'strength deck') (descriptive_representation_item.description = 'accomodation deck') (descriptive_representation_item.description = 'main deck') (descriptive_representation_item.description = 'weather deck') (descriptive_representation_item.description = 'navigation deck') (descriptive_representation_item.description = 'platform deck') (descriptive_representation_item.description = 'deck in superstructure') (descriptive_representation_item.description = 'bottom') (descriptive_representation_item.description = 'inner bottom') (descriptive_representation_item.description = 'double bottom') (descriptive_representation_item.description = 'outer shell')} </pre>

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'inner shell') (descriptive_representation_item.description = 'double shell') (descriptive_representation_item.description = 'bulkhead') (descriptive_representation_item.description = 'longitudinal bulkhead') (descriptive_representation_item.description = 'transversal bulkhead') (descriptive_representation_item.description = 'strength bulkhead') (descriptive_representation_item.description = 'wash bulkhead') (descriptive_representation_item.description = 'collision bulkhead') (descriptive_representation_item.description = 'hold bulkhead') (descriptive_representation_item.description = 'tank bulkhead') (descriptive_representation_item.description = 'wing bulkhead') (descriptive_representation_item.description = 'superstructure front bulkhead') (descriptive_representation_item.description = 'superstructure side bulkhead') (descriptive_representation_item.description = 'superstructure aft bulkhead') (descriptive_representation_item.description = 'stool') (descriptive_representation_item.description = 'wall') (descriptive_representation_item.description = 'transom') (descriptive_representation_item.description = 'sheer strake') (descriptive_representation_item.description =

Application Element	AIM element	Source	Rules	Reference Path
				'tank bottom') (descriptive_representation_item.description = 'tank side') (descriptive_representation_item.description = 'tank top') (descriptive_representation_item.description = 'hatch cover') (descriptive_representation_item.description = 'keel') (descriptive_representation_item.description = 'duct keel') (descriptive_representation_item.description = 'hatchway coaming') (descriptive_representation_item.description = 'hatchway sidecoaming') (descriptive_representation_item.description = 'hatchway endcoaming') (descriptive_representation_item.description = 'upper boom') (descriptive_representation_item.description = 'lower boom') (descriptive_representation_item.description = 'frame') (descriptive_representation_item.description = 'web frame') (descriptive_representation_item.description = 'vertical web frame') (descriptive_representation_item.description = 'transverse web frame') (descriptive_representation_item.description = 'floor') (descriptive_representation_item.description = 'transverse floor') (descriptive_representation_item.description = 'bracket') (descriptive_representation_item.description = 'user defined'))}
panel_system_functional_definition to panel_system (as defined_for)	PATH			extended_property_definition <= property_definition property_definition.definition ->

Application Element	AIM element	Source	Rules	Reference Path
				characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'panel system'>}
- from FUNCTIONAL_DEFINITION			387	
user_def_function	descriptive_representation_item. description	45		extended_property_definition <= property_definition = represented_definition <- PDR<'panel system function parameters'> -> representation {[representation.name = '.UNUSED.'] [representation.id = '.UNUSED.']} representation.items[i] -> DSC_REP_ITEM<'user def function'>
- from DEFINITION				
panel_system_functional_definition to unit (as local_units)	<i>see supertype</i>			
panel_system_functional_definition to global_id (as id)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PANEL_SYSTEM_PLANE_BOUNDARY	shape_aspect	41	388	{ROOT_CLASS<shape_aspect, 'panel system plane boundary'>}
panel_system_plane_boundary to plane (as plane)	PATH		365, 389	shape_aspect = represented_definition <- PDR<'panel system plane boundary design parameters'> -> representation representation.items[i] -> GEO_REP_ITEM<'plane', plane>
PANEL_SYSTEM_RELATIONSHIP	product_definition_relations hip	41		<i>see subtypes</i>
#1: panel_system_boundary_relationship				
panel_system_relationship to panel_system	product_definition_relations	41		{product_definition_relationship.relati

Application Element	AIM element	Source	Rules	Reference Path
(as item_1)	hip. relating_product_definition			on -> product_definition {CLASS_ID<product_definition, 'panel system'>}}
- from ITEM_RELATIONSHIP				
panel_system_relationship to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship.relating_product_definiti on -> product_definition [CLASS_ID<product_definition, 'panel system'>] [EXT_INST_REF<product_definition, 'ship structures', 'panel system'>]]}
panel_system_relationship to item (as item_2)	<i>see subtype</i>			
panel_system_relationship to external_instance_reference (as external_item_2)	<i>see subtype</i>			
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
panel_system_relationship to global_id (as id)	<i>see supertype</i>			
panel_system_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PLATE_STRAKE	[product_definition] [group]	41		{[LINK_TO_GROUP<product_definition> [CLASS<product_definition, 'plate strake', 'structural system'>] [CLASS<product_definition, 'structural system', 'system'>] [CLASS<product_definition, 'system', 'item structure'>] [CLASS<product_definition, 'item structure', 'definable object'>] [CLASS<product_definition, 'item structure', 'versionable object'>] [ROOT_CLASS<product_definition, 'versionable object'>] [CLASS<product_definition, 'system', 'item'>]

Application Element	AIM element	Source	Rules	Reference Path
				[CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]]
the_class	group.name	41		product_definition classification_item = product_definition classification_item <- applied_classification_assignment.items[i] applied_classification_assignment applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'class membership'} classification_assignment.assigned_classification -> group group.name {group.name = 'plate strake'}
- from STRUCTURAL_SYSTEM				
user_def_class	<i>see supertype</i>			
- from ITEM				
description	<i>see supertype</i>			
name	<i>see supertype</i>			
plate_strake to external_reference (as documentation)	<i>see supertype</i>			
plate_strake to ship (as ship_context)	<i>see supertype</i>			
- from ITEM_STRUCTURE				
plate_strake to external_instance_reference (as external_items)	PATH			group <- GROUPS<product_definition, 'item structure'> {[CLASS_ID<product_definition, 'plate'>] [EXT_INST_REF<product_definition, 'ship structures', 'plate'>]]}
plate_strake to external_instance_reference (as external_relationships)	PATH			group <- GROUPS<product_definition_relationship, 'item structure'>

Application Element	AIM element	Source	Rules	Reference Path
				{[CLASS_ID<product_definition_relationship, 'plate relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'plate relationship'>]]}
plate_strake to item (as items)	PATH			group <- GROUPS<product_definition, 'item structure'> {CLASS_ID<product_definition, 'plate'>}
plate_strake to item_relationship (as relationships)	PATH			group <- GROUPS<product_definition_relationship, 'item structure'> {CLASS_ID<product_definition, 'plate relationship'>}
- from DEFINABLE_OBJECT				
plate_strake to global_id (as id)	<i>see supertype</i>			
plate_strake to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
PLATE_STRAKE_FUNCTIONAL_DEFINITION	extended_property_definition	218	390	{[CLASS<extended_property_definition, 'plate strake functional definition', 'structural system functional definition'>] [CLASS<extended_property_definition, 'structural system functional definition', 'functional definition'>] [CLASS<extended_property_definition, 'functional definition', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_property_definition, 'versionable object'>]]}
the_function	descriptive_representation_item. description	45	362, 391	extended_property_definition <= property_definition = represented_definition <- PDR<plate strake function parameters> -> representation

Application Element	AIM element	Source	Rules	Reference Path
				<pre> {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] -> representation_item {representation_item.name = 'function'} representation_item => descriptive_representation_item {[(descriptive_representation_item.description = 'plating') (descriptive_representation_item.description = 'transom') (descriptive_representation_item.description = 'user defined')]]} </pre>
plate_strake_functional_definition to plate_strake (as defined_for)	PATH			<pre> extended_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'plate strake'>} </pre>
- from FUNCTIONAL_DEFINITION				
user_def_function	descriptive_representation_i tem. description	45	392	<pre> extended_property_definition <= property_definition = represented_definition <- PDR<'plate strake function parameters'> -> representation {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] -> DSC_REP_ITEM<'user def function'> </pre>
- from DEFINITION				
plate_strake_functional_definition to unit (as local_units)	<i>see supertype</i>			
plate_strake_functional_definition to global_id (as id)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
version_id	<i>see supertype</i>			
STRUCTURAL_SYSTEM	[product_definition] [group]	41		<pre> {[LINK_TO_GROUP<product_definition>] [CLASS<product_definition, 'structural\$system', 'system'>] [CLASS<product_definition, 'system', 'item\$structure'>] [CLASS<product_definition, 'item\$structure', 'definable\$object'>] [CLASS<product_definition, 'item\$structure', 'versionable\$object'>] [ROOT_CLASS<product_definition, 'versionable\$object'>] [CLASS<product_definition, 'system', 'item'>] [CLASS<product_definition, 'item', 'definable\$object'>] [ROOT_CLASS<product_definition, 'definable\$object'>] [(CLASS<product_definition, 'general\$structural\$system', 'structural\$system'>) (CLASS<product_definition, 'section', 'structural\$system'>) (CLASS<product_definition, 'panel system', 'structural system'>) (CLASS<product_definition, 'corrugated structure', 'structural system'>) (CLASS<product_definition, 'block', 'structural system'>) (CLASS<product_definition, 'plate strake', 'structural system'>) (CLASS<product_definition, 'built profile', 'structural system'>) (CLASS<product_definition, 'user defined', 'structural system'>)]}] </pre>
the_class	group.name	41		<pre> product_definition classification_item = product_definition classification_item <- applied_classification_assignment.items[i] </pre>

Application Element	AIM element	Source	Rules	Reference Path
				applied_classification_assignment applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'class membership'} classification_assignment.assigned_classification -> group group.name {(group.name = 'general structural system') (group.name = 'section') (group.name = 'panel system') (group.name = 'corrugated structure') (group.name = 'block') (group.name = 'plate stake') (group.name = 'built profile') (group.name = 'user defined')}
user_def_class	group.name	41	393	product_definition classification_item = product_definition classification_item <- applied_classification_assignment.items[i] applied_classification_assignment applied_classification_assignment <= classification_assignment {classification_assignment.role -> classification_role classification_role.name = 'user defined class'} classification_assignment.assigned_classification -> group group.name
- from ITEM				
description	product_definition.description	41		

Application Element	AIM element	Source	Rules	Reference Path
name	product_definition.name	41		
structural_system to external_reference (as documentation)	PATH			<pre> product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = 'UNUSED. ']} </pre>
structural_system to ship (as ship_context)	PATH	41		<pre> product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structure'} product_definition_formation.of_product -> product {CLASS_ID<product, 'ship'>} </pre>
- from ITEM_STRUCTURE				
structural_system to external_instance_reference (as external_items)	PATH			<pre> group <- GROUPS<product_definition, 'item structure'> {([CLASS_ID<product_definition, 'structural part'>] [EXT_INST_REF<product_definition, 'ship structures', 'structural part'>]) ([CLASS_ID<product_definition, 'structural system'>] [EXT_INST_REF<product_definition, 'ship structures', 'structural system'>])}] </pre>
structural_system to external_instance_reference (as external_relationships)	PATH			<pre> group <- GROUPS<product_definition_relationship, 'item structure'> {([CLASS_ID<product_definition_relationship, 'structural part relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural part relationship'>])}] </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> ([CLASS_ID<product_definition_relationship, 'structural system relationship'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural system relationship'>]]) </pre>
structural_system to item (as items)	PATH			<pre> group GROUPS<product_definition, 'item structure'> {((CLASS_ID<product_definition, 'structural part'>) (CLASS_ID<product_definition, 'structural system'>))} </pre>
structural_system to item_relationship (as relationships)	PATH			<pre> group <- GROUPS<product_definition_relationship, 'item structure'> {((CLASS_ID<product_definition, 'structural part relationship'>) (CLASS_ID<product_definition, 'structural system relationship'>))} </pre>
- from DEFINABLE_OBJECT				
structural_system to global_id (as id)	product_definition.id	41	359, 360	
structural_system to definition (as definitions)	PATH			<pre> product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition </pre>
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	361	<pre> ship_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version identifier'} </pre>
STRUCTURAL_SYSTEM_ADJACENCY_RELATIONSHIP	product_definition_relationship	41		<pre> {[CLASS<product_definition_relationship, 'structural system adjacency relationship', 'structural system relationship'>] [CLASS<product_definition_relationship, </pre>

Application Element	AIM element	Source	Rules	Reference Path
				'structural system relationship', 'system relationship'] [CLASS<product_definition_relationship, 'system relationship', 'item relationship'] [CLASS<product_definition_relationship, 'item relationship', 'definable object'] [ROOT_CLASS<product_definition_relationship, 'definable object'] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'] [ROOT_CLASS<product_definition_relationship, 'versionable object']]
boundary_index	shape_aspect	41		product_definition_relationship product_definition_relationship.related_product_definition -> PDCD<'structural system'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect {(CLASS_ID<shape_aspect, 'border'>) (EXT_INST_REF<shape_aspect, 'ship structures', 'border'>))}
displacement	value_representation_item. value_component	43	365, 394	product_definition_relationship product_definition_relationship.relatiing_product_definitio n -> PDCD <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect <- {CLASS_ID<shape_aspect, 'border'}

Application Element	AIM element	Source	Rules	Reference Path
				<pre> shape_aspect_relationship = represented_definition <- PDR<'structural system adjacency relationship design para meters'> -> representation {[representation.name = 'UNUSED.']} [representation.id = 'UNUSED.']} representation.items[i] -> representation_item {representation_item.name = 'displacement'} representation_item => value_representation_item {value_representation_item.value_component = length_measure} </pre>
structural_system_adjacency_relationship to structural_system (as item_2)	product_definition_relations hip. related_product_definition	41		<pre> {product_definition_relationship. related_product_definition -> PDCD<'structural system'> <- property_definition.definition property_definition => product_definition_shape product_definition_shape <- shape_aspect.of_shape shape_aspect <- {CLASS_ID<shape_aspect, 'border'>}} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {CLASS_ID<shape_aspect_relationship, 'structural system adjacency relationship'>}} </pre>
- from STRUCTURAL_SYSTEM_RELATIONSHIP				
structural_system_adjacency_relationship to structural_system (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				

Application Element	AIM element	Source	Rules	Reference Path
structural_system_adjacency_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
structural_system_adjacency_relationship to external_instance_reference (as external_item_2)	product_definition_relationship. related_product_definition	41		{product_definition_relationship.related_product_definition -> product_definition [CLASS_ID<product_definition, 'structural system'>] [EXT_INST_REF<product_definition, 'ship structures', 'structural system'>]}
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
structural_system_adjacency_relationship to global_id (as id)	<i>see supertype</i>			
structural_system_adjacency_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			
STRUCTURAL_SYSTEM_DESIGN_DEFINITION	extended_product_definition_shape	218	400	{[CLASS<extended_product_definition, 'structural system design definition', 'design definition'>] [CLASS<extended_product_definition, 'design definition', 'definition'>] [CLASS<extended_product_definition, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition, 'versionable object'>]}
tightness	descriptive_representation_item. description	41	363, 395	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'structural system design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'tightness'>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> { (descriptive_representation_item.description = 'unspecified') (descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'gas tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'user defined')} </pre>
user_defined_tightness	descriptive_representation_item. description	41	396	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'structural system design parameters'> -> representation representation.items[i] -> DSC_REP_ITEM<'user defined tightness'> </pre>
structural_system_design_definition to shape_representation (as representations)	PATH			<pre> extended_product_definition_shape <= product_definition_shape <= property_definition <- represented_definition <- PDR<'representation'> property_definition_representation.used_representation -> representation => shape_representation </pre>
structural_system_design_definition to structural_system (as defined_for)	PATH			<pre> extended_product_definition_shape <= product_definition_shape <= property_definition </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'structural system'>} </pre>
- from DEFINITION				
structural_system_design_definition to unit (as local_units)	PATH			<pre> extended_product_definition_shape <= product_definition_shape <= property_definition {property_definition.name = 'UNUSED.'} property_definition <- represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.context_of_items -> representation_context => global_unit_assigned_context {global_unit_assigned_context <= representation_context [representation_context.context_type = 'local units'] [representation_context.context_identifier = 'UNUSED.']} </pre>
structural_system_design_definition to global_id (as id)	extended_product_definition_shape. id	218	359, 364	
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment.	218	361	<pre> extended_product_definition_shape <- applied_identification_assignment.items[i] </pre>

Application Element	AIM element	Source	Rules	Reference Path
	assigned_id			applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
STRUCTURAL_SYSTEM_FUNCTIONAL_DEFINITION	extended_property_definition	218		<i>see subtypes</i>
#1: panel_system_functional_definition				
#2: plate_strake_functional_definition				
#3: corrugated_structure_functional_definition				
the_function	descriptive_representation_item. description	45		<i>see subtypes</i>
structural_system_functional_definition to structural_system (as defined_for)	PATH			<i>see subtypes</i>
- from FUNCTIONAL_DEFINITION				
user_def_function	descriptive_representation_item. description	45		<i>see subtypes</i>
- from DEFINITION				
structural_system_functional_definition to unit (as local_units)				
structural_system_functional_definition to global_id (as id)	extended_property_definition.id	218	359, 401	
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	361	extended_property_definition = identification_item <- applied_identification_assignment.items[I]

Application Element	AIM element	Source	Rules	Reference Path
				applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
STRUCTURAL_SYSTEM_RELATIONSHIP #1: structural_system_symmetry_relationship #2: panel_system_relationship #3: corrugated_structure_relationship	product_definition_relations hip	41		<i>see subtypes</i>
structural_system_relationship to structural_system (as item_1)	product_definition_relations hip. relating_product_definition	41		<i>see subtypes</i>
- from ITEM_RELATIONSHIP				
structural_system_relationship to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition	41		<i>see subtypes</i>
structural_system_relationship to item (as item_2)	<i>see subtypes</i>			
structural_system_relationship to external_instance_reference (as external_item_2)	<i>see subtypes</i>			
contxt	product_definition.descripti on	41		
- from DEFINABLE_OBJECT				
structural_system_relationship to global_id (as id)	product_definition_relations hip.id	41	359, 399	
structural_system_relationship to definition (as definitions)	PATH			product_definition_relationship = characterized_product_definition = characterized_definition <- property_definition.definition

Application Element	AIM element	Source	Rules	Reference Path
				property_definition
- from VERSIONABLE_OBJECT				
version_id	applied_identification_assignment. assigned_id	218	361	product_definition_relationship = identification_item <- applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
STRUCTURAL_SYSTEM_SYMMETRY_RELATIONSHIP	product_definition_relationship	41	366	{[CLASS<product_definition_relationship, 'structural system symmetry relationship', 'structural system relationship'>] [CLASS<product_definition_relationship, 'structural system relationship', 'system relationship'>] [CLASS<product_definition_relationship, 'system relationship', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]}
structural_system_symmetry_relationship to plane (as mirroring_plane)	PATH		365, 397	product_definition_relationship = {product_definition_relationship.name = 'UNUSED.' characterized_product_definition = characterized_definition <- property_definition.definition property_definition = {property_definition.name = 'UNUSED.' represented_definition <-

Application Element	AIM element	Source	Rules	Reference Path
				PDR<'structural system symmetry relationship design parameters'> -> representation representation.items[i] -> GEO_REPRESENTATION_ITEM<'mirroring plane', plane>
structural_system_symmetry_relationship to structural_system (as item_2)	product_definition_relations hip. related_product_definition	41		{product_definition_relationship.related_product_definition -> product_definition {CLASS_ID<product_definition, 'structural system'>}}
- from STRUCTURAL_SYSTEM_RELATIONSHIP				
structural_system_symmetry_relationship to structural_system (as item_1)	<i>see supertype</i>			
- from ITEM_RELATIONSHIP				
structural_system_symmetry_relationship to external_instance_reference (as external_item_1)	<i>see supertype</i>			
structural_system_symmetry_relationship to external_instance_reference (as external_item_2)	product_definition_relations hip. related_product_definition	41		{product_definition_relationship.related_product_definition -> product_definition [CLASS_ID<product_definition, 'structural system'>] [EXT_INST_REF<product_definition, 'ship structures', 'structural system'>]}
contxt	<i>see supertype</i>			
- from DEFINABLE_OBJECT				
structural_system_symmetry_relationship to global_id (as id)	<i>see supertype</i>			
structural_system_symmetry_relationship to definition (as definitions)	<i>see supertype</i>			
- from VERSIONABLE_OBJECT				
version_id	<i>see supertype</i>			

Table 17 - Mapping table for welds UoF

Application Element	AIM element	Source	Rules	Reference Path
BEVELED_GROOVE_WELD	extended_product_definition_shape	218	406	{[CLASS<extended_product_definition_shape, 'beveled groove weld', 'groove weld'>] [CLASS<extended_product_definition_shape, 'groove weld', 'welded joint design definition'>] [CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]}
endcut_shape	descriptive_representation_item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'endcut shape'> {(descriptive_representation_item.description = 'flare single V') (descriptive_representation_item.description = 'flare double V') (descriptive_representation_item.description = 'flare single bevel') (descriptive_representation_item.description = 'flare double bevel') (descriptive_representation_item.description = 'single

Application Element	AIM element	Source	Rules	Reference Path
				bevel') (descriptive_representation_item.description = 'double bevel') (descriptive_representation_item.description = 'single V') (descriptive_representation_item.description = 'double V') (descriptive_representation_item.description = 'single J') (descriptive_representation_item.description = 'double J') (descriptive_representation_item.description = 'single U') (descriptive_representation_item.description = 'double U') (descriptive_representation_item.description = 'user defined')
taper	descriptive_representation_ item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'taper' { (descriptive_representation_item.description = 'non taper') (descriptive_representation_item.description = 'one side taper') (descriptive_representation_item.description = 'both sides taper')}
taper_angle (OPT)	value_representation_item.v alue_-component	43	402, 431	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> VAL_REP_ITEM<'taper_angle', plane_angle_measure>

Application Element	AIM element	Source	Rules	Reference Path
- from GROOVE_WELD				
sidedness	descriptive_representation_ item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'sidedness'> {(descriptive_representation_item.description = 'one side') (descriptive_representation_item.description = 'both sides')}
weld_backing_type (OPT)	descriptive_representation_ item.description	45	431	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld backing type'> {(descriptive_representation_item.description = 'none') (descriptive_representation_item.description = 'permanent') (descriptive_representation_item.description = 'copper backing bar') (descriptive_representation_item.description = 'ceramic tape') (descriptive_representation_item.description = 'flare backing ring') (descriptive_representation_item.description = 'permanent backing ring') (descriptive_representation_item.description = 'removable backing ring')}

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'user defined')}
weld_joint_spacer	descriptive_representation_item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld joint spacer'> {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
- from WELDED_JOINT_DESIGN_-DEFINITION				
configuration	descriptive_representation_item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user

Application Element	AIM element	Source	Rules	Reference Path
				defined'})}
penetration	descriptive_representation_ item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')}
beveled_groove_weld to shape_-representation (as representations) (SET)	<i>See supertype</i>			
beveled_groove_weld to welded_joint (as defined_for) (SET [1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <= product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>}
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_ item.description</i>	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				DSC_REP_ITEM<'joint orientation'> {((descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge'))}
tightness	descriptive_representation_ item.description	45	408	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'beveled groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {((descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined'))}
- from DEFINITION				
beveled_groove_weld to global_id (as id)	PATH			<i>See supertype</i>
beveled_groove_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
BUTT_GROOVE_WELD	extended_product_definition_shape	218	410	<pre> {[CLASS<extended_product_definition_shape, 'butt groove weld', 'groove weld'>] [CLASS<extended_product_definition_shape, 'groove weld', 'welded joint design definition'>] [CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]}</pre>
face_shape	descriptive_representation_item.description	45	412	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'butt groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'face shape'> {(descriptive_representation_item.description = 'square') (descriptive_representation_item.description = 'scarf') (descriptive_representation_item.description = 'user defined')}</pre>
- from GROOVE_WELD				
sidedness	descriptive_representation_item.description	45	412	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'butt groove weld design parameters'> -></pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> representation representation.items [i] -> DSC_REP_ITEM<'sidedness'> {(descriptive_representation_item.description = 'one side') (descriptive_representation_item.description = 'both sides')} </pre>
weld_backing_type (OPT)	descriptive_representation_ - item.description	45	433	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'butt groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld backing type'> {(descriptive_representation_item.description = 'none') (descriptive_representation_item.description = 'permanent') (descriptive_representation_item.description = 'copper backing bar') (descriptive_representation_item.description = 'ceramic tape') (descriptive_representation_item.description = 'flare backing ring') (descriptive_representation_item.description = 'permanent backing ring') (descriptive_representation_item.description = 'removable backing ring') (descriptive_representation_item.description = 'user defined')} </pre>
weld_joint_spacer	descriptive_representation_ - item.description	45	412	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- </pre>

Application Element	AIM element	Source	Rules	Reference Path
				PDR<'butt groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld joint spacer'> {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
- from WELDED_JOINT_DESIGN_-DEFINITION				
configuration	descriptive_representation_ item.description	45	412	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'butt groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user defined')}
penetration	descriptive_representation_ item.description	45	412	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'butt groove weld design parameters'> ->

Application Element	AIM element	Source	Rules	Reference Path
				representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')}
butt_groove_weld to shape_representation (as representations) (SET)	<i>See supertype</i>			
butt_groove_weld to welded_joint (as defined_for) (SET [1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <= product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>}
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_- item.description</i>	45	412	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'butt groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint orientation'> {(descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge')}

Application Element	AIM element	Source	Rules	Reference Path
tightness	descriptive_representation_ item.description	45	412	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'butt groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {(descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined')}}
- from DEFINITION				
butt_groove_weld to global_id (as id)	PATH			<i>See supertype</i>
butt_groove_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			
CONTINUOUS_FILLET_WELD	extended_product_definition_ _shape	218	415	{[CLASS<extended_product_definition_shape, 'continuous fillet weld', 'fillet weld'>] [CLASS<extended_product_definition_shape, 'fillet weld', 'welded joint design definition'>] [CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design

Application Element	AIM element	Source	Rules	Reference Path
				definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]
- from FILLET_WELD				
endcut_shape_type	descriptive_representation_item.-description	45	435	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'continuous fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'endcut shape type'> {(descriptive_representation_item.description = 'fillet V') (descriptive_representation_item.description = 'fillet J') (descriptive_representation_item.description = 'user defined')}
sidedness	descriptive_representation_item.-description	45	435	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'continuous fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'sidedness'> {(descriptive_representation_item.description = 'one side') (descriptive_representation_item.description = 'both sides')}

Application Element	AIM element	Source	Rules	Reference Path
- from WELDED_JOINT_DESIGN_-DEFINITION				
configuration	descriptive_representation_- item.description	45	435	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'continuous fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user defined')}
penetration	descriptive_representation_- item.description	45	435	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'continuous fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')}
contonuous_fillet_weld to shape_-	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
representation (as representations) (SET)				
contonuous_fillet_weld to welded_joint (as defined_for) (SET [1:?])	PATH			<pre> extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>} </pre>
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_- item.description</i>	45	435	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'continuous fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint orientation'> {(descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge')} </pre>
tightness	descriptive_representation_- item.description	45	435	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'continuous fillet weld design parameters'> -> </pre>

Application Element	AIM element	Source	Rules	Reference Path
				<pre> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> { (descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined')} </pre>
- from DEFINITION				
contonuous_fillet_weld to global_id (as id)	PATH			<i>See supertype</i>
contonuous_fillet_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			
ELECTRODE_CHEMICAL_-COMPOSITION	representation	43		{ROOT_CLASS<representation, 'electrode chemical omposition'>}
aluminium	value_representation_item.v alue_-component	43	402, 416	<pre> representation representation.items [i] -> VAL_REP_ITEM<'aluminium', ratio_measure> </pre>
carbon	value_representation_item.v alue_-component	43	402, 416	<pre> representation representation.items [i] -> VAL_REP_ITEM<'carbon', ratio_measure> </pre>
chromium	value_representation_item.v alue_-component	43	402, 416	<pre> representation representation.items [i] -> VAL_REP_ITEM<'chromium', ratio_measure> </pre>

Application Element	AIM element	Source	Rules	Reference Path
copper	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'copper', ratio_measure>
manganese	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'manganese', ratio_measure>
molybdenum	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'molybdenum', ratio_measure>
nickel	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'nickel', ratio_measure>
Phosphorus	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'phosphorus', ratio_measure>
silicon	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'silicon', ratio_measure>
sulphur	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'sulphur', ratio_measure>
titanium	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'titanium', ratio_measure>
vanadium	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'vanadium', ratio_measure>
zirconium	value_representation_item.v alue_-component	43	402, 416	representation representation.items [i] -> VAL_REP_ITEM<'zirconium', ratio_measure>
FILLET_WELD	extended_product_definition _shape	41	414	[CLASS<extended_product_definition_shape, 'fillet weld', 'welded joint design definition'] [CLASS<extended_product_definition_shape, 'welded

Application Element	AIM element	Source	Rules	Reference Path
				joint design definition', 'structural part joint design definition'] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'] [CLASS<extended_product_definition_shape, 'design definition', 'definition'] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'] [ROOT_CLASS<extended_product_definition_shape, 'versionable object']
endcut_shape_type	descriptive_representation_item.description	45	417	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'endcut shape type'> {(descriptive_representation_item.description = 'fillet V') (descriptive_representation_item.description = 'fillet J') (descriptive_representation_item.description = 'user defined')}
sidedness	descriptive_representation_item.description	45	417	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'sidedness'> {(descriptive_representation_item.description = 'one side') (descriptive_representation_item.description = 'both sides')}

Application Element	AIM element	Source	Rules	Reference Path
- from WELDED_JOINT_DESIGN_DEFINITION				
configuration	descriptive_representation_item.description	45	417	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user defined')} </pre>
penetration	descriptive_representation_item.description	45	417	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')} </pre>

Application Element	AIM element	Source	Rules	Reference Path
fillet_weld to welded_joint (as defined_for) (SET [1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>}
fillet_weld to shape_representation (as representations) (SET)	<i>See supertype</i>			
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_- item.description</i>	45	417	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint orientation'> {(descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge')}
tightness	descriptive_representation_- item.description	45	417	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'fillet weld design parameters'> ->

Application Element	AIM element	Source	Rules	Reference Path
				<pre> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> { (descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined') }</pre>
- from DEFINITION				
fillet_weld to global_id (as id)	PATH			<i>See supertype</i>
fillet_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			
GROOVE_WELD	extended_product_definition_shape	218	418	<pre> [CLASS<extended_product_definition_shape, 'groove weld', 'welded joint design definition'>] [CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>]]] [ROOT_CLASS<extended_product_definition_shape,</pre>

Application Element	AIM element	Source	Rules	Reference Path
				'versionable object']}]
sidedness	descriptive_representation_ item.description	45	419	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'sidedness'> {(descriptive_representation_item.description = 'one side') (descriptive_representation_item.description = 'both sides')}
weld_backing_type (OPT)	descriptive_representation_ item.description	45	430	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld backing type'> {(descriptive_representation_item.description = 'none') (descriptive_representation_item.description = 'permanent') (descriptive_representation_item.description = 'copper backing bar') (descriptive_representation_item.description = 'ceramic tape') (descriptive_representation_item.description = 'flare backing ring') (descriptive_representation_item.description = 'permanent backing ring') (descriptive_representation_item.description = 'removable backing ring')}

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'user defined')}
weld_joint_spacer	descriptive_representation_item.description	45	419	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld joint spacer'> {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
- from WELDED_JOINT_DESIGN_-DEFINITION				
configuration	descriptive_representation_item.description	45	419	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user

Application Element	AIM element	Source	Rules	Reference Path
				defined')}
penetration	descriptive_representation_ item.description	45	419	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')}
groove_weld to welded_joint (as defined_for) (SET [1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <= product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>}
groove_weld to shape_representation (as representations) (SET)	<i>See supertype</i>			
- from STRUCTURAL_PART_JOINT_ DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_ item.description</i>	45	419	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				DSC_REP_ITEM<'joint orientation'> {((descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge'))}
tightness	descriptive_representation_ - item.description	45	419	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'groove weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {((descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined'))}
- from DEFINITION				
groove_weld to global_id (as id)	PATH			<i>See supertype</i>
groove_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
INTERMITTENT_FILLET_WELD	extended_product_definition_shape	218	420	{[CLASS<extended_product_definition_shape, 'intermittent fillet weld', 'fillet weld'>] [CLASS<extended_product_definition_shape, 'fillet weld', 'welded joint design definition'>] [CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]}
cutout_rules	descriptive_representation_item.description	45	421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'cutout rules'> {(descriptive_representation_item.description = 'none') (descriptive_representation_item.description = 'fixed rule') (descriptive_representation_item.description = 'member depth') (descriptive_representation_item.description = 'percent length')}
end_rules	descriptive_representation_item.description	45	421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <=

Application Element	AIM element	Source	Rules	Reference Path
				PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'end rules'> {(descriptive_representation_item.description = 'none') (descriptive_representation_item.description = 'fixed rule') (descriptive_representation_item.description = 'member depth') (descriptive_representation_item.description = 'percent length')}
fillet_alignment	descriptive_representation_ - item.description	45	421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'fillet alignment'> {(descriptive_representation_item.description = 'staggered') (descriptive_representation_item.description = 'chained')}
fillet_weld_length	value_representation_item.v alue_-component	43	402, 421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> VAL_REP_ITEM<'fillet weld length', length_measure>
fillet_weld_spacing	value_representation_item.v alue_-component	43	402, 421	extended_product_definition_shape <= product_definition_shape <= property_definition =

Application Element	AIM element	Source	Rules	Reference Path
				<pre> represented_definition <- PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> VAL_REP_ITEM<'fillet weld spacing', length_measure> </pre>
penetration_rules	descriptive_representation_item.-description	45	421	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration rules'> {(descriptive_representation_item.description = 'none') (descriptive_representation_item.description = 'fixed rule')} (descriptive_representation_item.description = 'member depth') (descriptive_representation_item.description = 'percent length')} </pre>
- from FILLET_WELD				
endcut_shape_type	descriptive_representation_item.-description	45	421	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'endcut shape type'> {(descriptive_representation_item.description = 'fillet V') (descriptive_representation_item.description = 'fillet J') (descriptive_representation_item.description = 'user defined')} </pre>

Application Element	AIM element	Source	Rules	Reference Path
sidedness	descriptive_representation_item.-description	45	421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'sidedness'> {(descriptive_representation_item.description = 'one side') (descriptive_representation_item.description = 'both sides')}
- from WELDED_JOINT_DESIGN_-DEFINITION				
configuration	descriptive_representation_item.description	45	421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'intermittent fillet weld parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user defined')}

Application Element	AIM element	Source	Rules	Reference Path
penetration	descriptive_representation_- item.description	45	421	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')} </pre>
intermittent_fillet_weld to welded_joint (as defined_for) (SET [1:?])	PATH			<pre> extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>} </pre>
intermittent_fillet_weld to shape_- representation (as representations) (SET)	<i>See supertype</i>			
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_- item.description</i>	45	421	<pre> extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> </pre>

Application Element	AIM element	Source	Rules	Reference Path
				DSC_REP_ITEM<'joint orientation'> {((descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge'))}
tightness	descriptive_representation_ - item.description	45	421	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'intermittent fillet weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {((descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined'))}
- from DEFINITION				
intermittent_fillet_weld to global_id (as id)	PATH			<i>See supertype</i>
intermittent_fillet_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			

Application Element	AIM element	Source	Rules	Reference Path
SPOT_SEAM_WELD	extended_product_definition_shape	218	422422 422	{[CLASS<extended_product_definition_shape, 'spot seam weld', 'welded joint design definition'>] [CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape, 'versionable object'>]]}
weld_context	descriptive_representation_item.description	45	423	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'spot seam weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'weld context'> {(descriptive_representation_item.description = 'spot') (descriptive_representation_item.description = 'seam')}
- from WELDED_JOINT_DESIGN_DEFINITION				
configuration	descriptive_representation_item.description	45	423	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'spot seam weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt')}

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user defined')]]}
penetration	descriptive_representation_ item.description	45	423	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'spot seam weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')}
spot_seam_weld to welded_joint (as defined_for) (SET [1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <= product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>}

Application Element	AIM element	Source	Rules	Reference Path
spot_seam_weld to shape_representation (as representations) (SET)	<i>See supertype</i>			
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_- item.description</i>	45	423	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'spot seam weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint orientation'> {(descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge')}
tightness	descriptive_representation_- item.description	45	423	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'spot seam weld design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {(descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight')}

Application Element	AIM element	Source	Rules	Reference Path
				(descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined')}
- from DEFINITION				
spot_seam_weld to global_id (as id)	PATH			<i>See supertype</i>
spot_seam_weld to unit (as local_units) (SET) (S[0:0])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			
STRUCTURAL_PART_CONNECTION_- IMPLEMENTATION	product_definition	41		{[CLASS<product_definition, 'structural part connection implementation', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]]}
structural_part_connection_implementation to structural_part_joint (as realization_of)	PATH			product_definition <- product_definition_relationship.relatiing_- product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'structural part joint'>}
- from ITEM				
name	product_definition.name	41		
structural_part_connection_implementation to external_reference (as documentation) (SET) (S[0:?])	PATH			product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = '.UNUSED. ']}
description (OPT)	product_definition.descripti	41		

Application Element	AIM element	Source	Rules	Reference Path
(OPT)	on			
structural_part_connection_-implementation to ship (as ship_context) (OPT)	PATH			product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structure'} {product_definition_formation.description = 'ship context'} product_definition_formation.of_product -> product {CLASS_ID<product, 'ship'>}
- from DEFINABLE_OBJECT				
structural_part_connection_-implementation to global_id (as id)	product_definition.id	41	403, 404	
structural_part_connection_-implementation to definition (as definitions) (SET) (INV)	PATH			product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition
STRUCTURAL_PART_JOINT	product_definition_relations hip	41		{[CLASS<product_definition_relationship, 'structural part joint', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object'>] [ROOT_CLASS<product_definition_relationship, 'definable object'>] [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]}
joint_type	descriptive_representation_- item.description	45		product_definition_relationship product_definition_relationship.relatng_product_- definition -> product_definition

Application Element	AIM element	Source	Rules	Reference Path
				PDCD <- property_definition.definition property_definition = represented_definition <- PDR<'structural part joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint type'> {(descriptive_representation_item.description = 'fixed joint') (product_definition.description = 'movable joint'))}
structural_part_joint to structural_part_connection_implementation (as realization) (SET[1:?])	PATH			product_definition_relationship {[product_definition_relationship.name = '.UNUSED.'] [product_definition_relationship.description = 'realization']} product_definition_relationship.relatng_product_ definition -> product_definition {CLASS_ID<product_definition, 'structural part connection implementation'>}}
- from ITEM_RELATIONSHIP				
structural_part_joint to item (as item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship. relating_product_definition -> product_definition {CLASS_ID<product_definition, 'structural part'>}}
structural_part_joint to external_instance_reference (as external_item_1)	product_definition_relations hip. relating_product_definition	41		{product_definition_relationship. relating_product_definition -> product_definition [CLASS_ID<product_definition, 'structural part'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural part joint'>]}
structural_part_joint to item (as item_2)	product_definition_relations hip.	41		{product_definition_relationship. relating_product_definition ->

Application Element	AIM element	Source	Rules	Reference Path
	related_product_definition			product_definition {CLASS_ID<product_definition, 'structural part'>}}
structural_part_joint to external_instance_reference (as external_item_2)	product_definition_relations hip. related_product_definition	41		{product_definition_relationship. relating_product_definition -> product_definition [CLASS_ID<product_definition, 'structural part'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'structural part joint'>]}
contxt	<i>See supertype</i>			
- from DEFINABLE_OBJECT				
structural_part_joint to definition (as definitions) (SET) (INV)	PATH			product_definition_relationship {[product_definition_relationship.name = '.UNUSED.'] [product_definition_relationship.description = 'definitions']} product_definition_relationship.relating_product_ definition -> product_definition = characterized_product_definition = characterized_definition <- property_definition.definition property_definition
structural_part_joint to global_id (as id)	product_definition_relations hip.id	41	403, 407	
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	405	VERSION_ID<product_definition_relationship>
STRUCTURAL_PART_JOINT_ DESIGN_DEFINITION	extended_product_definition_s hape	218	432	{[CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>] [ROOT_CLASS<extended_product_definition_shape,

Application Element	AIM element	Source	Rules	Reference Path
				'versionable object'>]
joint_orientation	descriptive_representation_ item.description	45	434	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'structural part joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint orientation'> {(descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge edge')}]
tightness	descriptive_representation_ item.description	45	434	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'structural part joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {(descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description =

Application Element	AIM element	Source	Rules	Reference Path
				'undefined'))}
structural_part_joint_design_definition to structural_part_joint (as defined_for) (SET[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition <= property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <= product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'structural part joint'>}
- from DEFINITION				
structural_part_joint_design_definition to global_id (as id)	extended_product_definition _shape.id		403, 411	
structural_part_joint_design_definition to unit (as local_units) (SET) (S[0:?])	PATH			<i>See supertype</i>
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	405	VERSION_ID<extended_product_definition_shape>
WELD	product_definition	41		{[CLASS<product_definition, 'weld', 'structural part connection implementation'>] [CLASS<product_definition, 'structural part connection implementation', 'item'>] [CLASS<product_definition, 'item', 'definable object'>] [ROOT_CLASS<product_definition, 'definable object'>]]}
weld to welded_joint (as realization_of)	PATH			{[product_definition.name = 'UNUSED.'] [product_definition.description = 'realization of'] product_definition <= product_definition_relationship.related_product_definition

Application Element	AIM element	Source	Rules	Reference Path
				product_definition_relationship {[CLASS_ID<product_definition_relationship 'welded joint'>]}
- from STRUCTURAL_PART_- CONNECTION_IMPLEMENTATION				
weld to structural_part_joint (as realization_of)	PATH			{[product_definition.name = '.UNUSED.' [product_definition.description = 'realization of' product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {[CLASS_ID<product_definition_relationship 'structural part joint'>}]}
- from ITEM				
name	product_definition.name	41		
weld to external_reference (as documentation) (S[0:?])	PATH			product_definition = document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> object_role [object_role.name = 'documentation'] [object_role.description = '.UNUSED. ']}
description (OPT)	product_definition.description	41		
weld to ship (as ship_context)	PATH			product_definition product_definition.formation -> product_definition_formation {product_definition_formation.id = 'ship structure'} product_definition_formation.of_product -> product {CLASS_ID<product, 'ship'>}
- from DEFINABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
weld to definition (as definitions) (SET) (INV)	PATH			<i>See supertype</i>
weld to global_id (as id)	product_definition.id		403, 404	
WELD_DESIGN_DEFINITION	representation	43		{[CLASS<representation, 'weld design definition', 'design definition'>] [CLASS<representation, 'design definition', 'definition'>] [CLASS<representation, 'definition', 'versionable object'>] [ROOT_CLASS<representation, 'versionable object'>]}
connection_angle	value_representation_item.v alue_-component	43	402, 424	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'connection angle', plane_angle_measure>
end_suspension (OPT)	value_representation_item.v alue_-component	43	402, 429	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'end suspension', length_measure>
inclination_angle	value_representation_item.v alue_-component	43	402, 424	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'inclination angle', plane_angle_measure>
rotation_angle	value_representation_item.v alue_-component	43	402, 424	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'rotation angle', plane_angle_measure>
shape_of_weld_surface	descriptive_representation_i tem.-description	45	424	representation representation.items [i] -> DSC_REP_ITEM<'shape of weld surface'>

Application Element	AIM element	Source	Rules	Reference Path
				{(descriptive_representation_item.description = 'flush') (descriptive_representation_item.description = 'convex') (descriptive_representation_item.description = 'concave') (descriptive_representation_item.description = 'none')}
shrinkage	value_representation_item.v alue_-component	43	402, 424	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'shrinkage', length_measure>
start_suspension (OPT)	value_representation_item.v alue_-component	43	402, 429	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'start suspension', length_measure>
weld_cross_section_area (OPT)	value_representation_item.v alue_-component	43	402, 429	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'weld cross section area', area_measure>
weld_size (SET)	value_representation_item.v alue_-component	43	402, 424	representation representation.items [i] -> representation_item => VAL_REP_ITEM<'weld size', length_measure>
weld_design_definition to curve (as weld_geometry)	PATH			representation representation.items [i] -> representation_item {representation_item.name = 'weld geometry'} representation_item => geometric_representation_item => curve
weld_design_definition to weld (as defined_for) (SET[1:?])	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition =

Application Element	AIM element	Source	Rules	Reference Path
				product_definition {CLASS_ID<product_definition, 'weld'>}
- from DEFINITION				
weld_design_definition to global_id (as id)	<i>representation.id</i>		403, 409	
weld_design_definition to unit (as local_units) (SET) (S[0:?])	PATH			representation LOCAL_UNITS
- from VERSIONABLE_OBJECT				
version_id	identification_assignment. assigned_id	41	405	VERSION_ID<representation>
WELD_FILLER_MATERIAL	extended_property_definition	218	425	{[CLASS<extended_property_definition, 'weld filler material', 'ship material property'>] [CLASS<extended_property_definition, 'ship material property', 'definition'>] [CLASS<extended_property_definition, 'definition', 'versionable object' [ROOT_CLASS<extended_property_definition, 'versionable object'>]]}
chemical_composition_id (OPT)	descriptive_representation_ item.description	45	426	extended_property_definition <= property_definition = represented_definition <- PDR<'weld filler material design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'chemical composition identifier'>
electrode_id (OPT)	descriptive_representation_ item.description	45	426	extended_property_definition <= property_definition = represented_definition <- PDR<'weld filler material design parameters'> -> representation representation.items [i] ->

Application Element	AIM element	Source	Rules	Reference Path
				DSC_REP_ITEM<'electrode identifier'>
material_strength_id (OPT)	descriptive_representation_ item.description	45	426	extended_property_definition <= property_definition = represented_definition <= PDR<'weld filler material design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'material strength identifier'>
notch_impact_work_id (OPT)	descriptive_representation_ item.description	45	426	extended_property_definition <= property_definition = represented_definition <= PDR<'weld filler material design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'notch impact work identifier'>
weld_filler_material to electrode_chemical_composition (as chemical_composition)	PATH			extended_property_definition <= property_definition = represented_definition <= PDR<'weld filler material design parameters'> -> representation {representation.items [i] -> representation_item representation_item.name = 'chemical composition'} {CLASS_ID<representation, 'electrode chemical composition'>}
weld_filler_material to weld (as defined_for) (SET[1:?])	PATH			extended_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {CLASS_ID<product_definition, 'weld'>}
- from SHIP_MATERIAL_PROPERTY				

Application Element	AIM element	Source	Rules	Reference Path
description (OPT)	property_definition.description	41		extended_property_definition <= property_definition property_definition.description
density	<i>See supertype</i>			
weld_filler_material to document_ reference_with_address (as material_reference) (OPT)	PATH			extended_property_definition = documentation_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference { document_reference.role -> object_role object_role.name = 'material reference'}
weld_filler_material to item (as defined_for) (SET[1:?]) (S[1:?])	PATH			extended_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition {(CLASS_ID<product_definition, 'structural part'>) (CLASS_ID<product_definition, 'weld'>)}
- from DEFINITION				
weld_filler_material to global_id (as id)	<i>extended_property_definition.i d</i>	218	403, 413	
weld_filler_material to unit (as local_units) (SET) (S[0:?])	PATH			extended_property_definition <= property_definition {property_definition.name = 'UNUSED.'} property_definition = represented_definition <- PDR<'weld filler material design parameters'> -> representation LOCAL_UNITS
- from VERSIONABLE_OBJECT				

Application Element	AIM element	Source	Rules	Reference Path
version_id	identification_assignment. assigned_id	41	405	VERSION_ID<extended_property_definition>
WELDED_JOINT	product_definition_relations hip	41		{[CLASS<product_definition_relationship, 'welded joint', 'structural part joint'>] [CLASS<product_definition_relationship, 'structural part joint', 'item relationship'>] [CLASS<product_definition_relationship, 'item relationship', 'definable object' [ROOT_CLASS<product_definition_relationship, 'definable object' [CLASS<product_definition_relationship, 'item relationship', 'versionable object'>] [ROOT_CLASS<product_definition_relationship, 'versionable object'>]]]}
joint_type	descriptive_representation_ item.description	45		product_definition_relationship product_definition_relationship.relating_product_definitio n -> PDCD <- property_definition.definition property_definition = represented_definition <- PDR<'welded joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint type'> {descriptive_representation_item.description = 'fixed joint'}
welded_joint to weld (as realization) (SET)	PATH			product_definition_relationship product_definition_relationship.relating_product_- definition -> product_definition {CLASS_ID<product_definition, 'weld '>}

Application Element	AIM element	Source	Rules	Reference Path
- from STRUCTURAL_PART_JOINT				
joint_type	descriptive_representation_item.description	45		<pre> product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition PDCD <- property_definition.definition property_definition = represented_definition <- PDR<'structural part joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint type'> {descriptive_representation_item.description = 'fixed joint'}</pre>
structural_part_joint to structural_part_connection_implementation (as realization) (SET[1:?])	PATH			<pre> product_definition_relationship {[product_definition_relationship.name = '.UNUSED.']} [product_definition_relationship.description = 'realization']} product_definition_relationship.relating_product_definition -> product_definition {CLASS_ID<product_definition, 'structural part connection implementation'>}</pre>
- from ITEM_RELATIONSHIP				
contxt	<i>See supertype</i>			
welded_joint to external_instance_reference (as external_item_1)	product_definition_relationship. relating_product_definition			<pre> {product_definition_relationship. relating_product_definition -> product_definition [CLASS_ID<product_definition, 'structural part'>] [EXT_INST_REF<product_definition_relationship,</pre>

Application Element	AIM element	Source	Rules	Reference Path
				'ship structures', 'welded joint'>}}
welded_joint to external_instance_-reference (as external_item_2)	product_definition_relations hip. related_product_definition			{product_definition_relationship. relating_product_definition -> product_definition [CLASS_ID<product_definition, 'structural part'>] [EXT_INST_REF<product_definition_relationship, 'ship structures', 'welded joint'>}]}
welded_joint to item (as item_1)	product_definition_relations hip. relating_product_definition			{product_definition_relationship. relating_product_definition -> product_definition {CLASS_ID<product_definition, 'structural part'>}}
welded_joint to item (as item_2)	product_definition_relations hip. related_product_definition			{product_definition_relationship. relating_product_definition -> product_definition {CLASS_ID<product_definition, 'structural part'>}}
- from DEFINABLE_OBJECT				
welded_joint to global_id (as id)	product_definition_relations hip.id		403, 407	
welded_joint to definition (as definitions) (SET) (INV)	PATH			See supertype
- from VERSIONABLE_OBJECT				
version_id	See supertype			
WELDED_JOINT_DESIGN_DEFINITION	extended_product_definition _shape	218	427	{[CLASS<extended_product_definition_shape, 'welded joint design definition', 'structural part joint design definition'>] [CLASS<extended_product_definition_shape, 'structural part joint design definition', 'design definition'>] [CLASS<extended_product_definition_shape, 'design definition', 'definition'>] [CLASS<extended_product_definition_shape, 'definition', 'versionable object'>]}

Application Element	AIM element	Source	Rules	Reference Path
				[ROOT_CLASS<extended_product_definition_shape, 'versionable object' >]]
configuration	descriptive_representation_ item.description	45	428	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'welded joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'configuration'> {(descriptive_representation_item.description = 'butt') (descriptive_representation_item.description = 'tee') (descriptive_representation_item.description = 'corner') (descriptive_representation_item.description = 'lap') (descriptive_representation_item.description = 'cruciform') (descriptive_representation_item.description = 'edge') (descriptive_representation_item.description = 'stud') (descriptive_representation_item.description = 'surfacing') (descriptive_representation_item.description = 'flare groove') (descriptive_representation_item.description = 'user defined')}
penetration	descriptive_representation_ item.description	45	428	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'welded joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'penetration'> {(descriptive_representation_item.description = 'full') (descriptive_representation_item.description = 'partial')}

Application Element	AIM element	Source	Rules	Reference Path
welded_joint_design_definition to shape_representation (as representations) (SET)	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'welded joint design parameters'> -> property_definition_representation.used_representation -> representation => shape_representation
welded_joint_design_definition to welded_joint (as defined_for) (SET[1:?])	PATH			extended_product_definition_shape <= product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition = product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {CLASS_ID<product_definition_relationship, 'welded joint'>}
- from STRUCTURAL_PART_JOINT_- DESIGN_DEFINITION				
joint_orientation	<i>descriptive_representation_- item.description</i>	45	428	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <- PDR<'welded joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'joint orientation'> {(descriptive_representation_item.description = 'face face') (descriptive_representation_item.description = 'face edge') (descriptive_representation_item.description = 'edge

Application Element	AIM element	Source	Rules	Reference Path
				edge'})}
tightness	descriptive_representation_ item.description	45	428	extended_product_definition_shape <= product_definition_shape <= property_definition = represented_definition <= PDR<'weld joint design parameters'> -> representation representation.items [i] -> DSC_REP_ITEM<'tightness'> {(descriptive_representation_item.description = 'non tight') (descriptive_representation_item.description = 'air tight') (descriptive_representation_item.description = 'fume tight') (descriptive_representation_item.description = 'oil tight') (descriptive_representation_item.description = 'weather tight') (descriptive_representation_item.description = 'water tight') (descriptive_representation_item.description = 'undefined')}
- from DEFINITION			428	
welded_joint_design_definition to global_id (as id)	extended_product_definition _shape.id		403, 411	
welded_joint_design_definition to unit (as local_units) (S[0:0])	PATH			See supertype
- from VERSIONABLE_OBJECT				
version_id	<i>See supertype</i>			

Global rules:

- 1 A class shall never participate in a group_assignment.
- 2 RJ36<applied_approval_assignment, items, 1, 'subject'>: The number of items in an applied_approval_assignment that plays the role of 'subject' (of an approval) shall always be exactly one.
- 3 R61<applied_approval_assignment, 'subject', assigned_approval, 'approvals', 'approval_history', approval>: All applied_approval_assignments with role 'subject' that via assigned_approval reference approvals that are in an applied_group_assignment with role 'approvals' and with a group of class 'approval_history' shall reference the same one instance in the set of 'items' of the applied_approval_assignments.
- 4 R3: ensures that the representation has a global_unit_assigned_context.
- 5 Approval_assignments with role 'proposed alternative' shall reference via assigned_approval that approval instance which is the reason why an alternative is proposed; i.e. both these related approval_assignments (with roles 'subject' and 'proposed alternative' respectively) shall reference the same approval instance.
- 6 RJ38<group, 'structural class rejection', 1, applied_document_reference, items, 'explanations'>: A group of class 'structural class rejection' shall be a member within at least one applied_document_reference.items with object_role.name of the applied_document_reference equal 'explanations'.
- 7 R10<'plate renewal definition attributes', 'rule thickness', 'renewal thickness'>: forces the items of the representation that is referenced by a property_definition_representation with name 'plate renewal definition attributes' to have maximum the 2 listed elements.
- 8 R10<'profile renewal definition attributes', 'renewal modulus', 'rule inertia', 'rule section modulus'>: forces the items of the representation that is referenced by a property_definition_representation with name 'profile renewal definition attributes' to have maximum the 3 listed elements.
- 9 RJ37<'hull cross section result', 'actual first moment horizontal', 'actual first moment vertical', 'actual inertia cross', 'actual inertia horizontal', 'actual inertia torsion', 'actual inertia vertical', 'actual neutral axis position horizontal', 'actual neutral axis position vertical', 'actual section area', 'actual section modulus bottom', 'actual section modulus coaming', 'actual section modulus deck', 'actual stress bottom', 'actual stress coaming', 'actual stress deck', 'actual twist centre horizontal', 'actual twist centre vertical', 'direction largest inertia', 'direction least inertia', 'max moment', 'max shear force', 'meter weight', 'min moment', 'min shear force', 'required section modulus bottom', 'required section modulus coaming', 'required section modulus deck', 'required stress bottom', 'required stress coaming', 'required stress deck'>: forces the items list of a representation of class 'hull cross section result' to include exactly one instance of each representation_item with the names that are provided in the argument list (30 names !).
- 10 R58<applied_group_assignment, items, 1, 'approvals', approval>: An applied_group_assignment with object_role.name = 'approvals' shall have only approval entities in its list of items and at least one of those.
- 11 Ensure uniqueness of APPROVAL_HISTORY.approvals.
- 12 If role="subject", then the list must contain exactly 1 item.
- 13 R8 This group contains a single element only, since there may be only one definable_object as context.
- 14 Each approval_assignment shall have only one approval_item, which is the image of definition. (see class_approvals).
- 15 This OPTIONAL attribute must be present.
- 16 One element required at minimum.
- 17 similar to R19<> CARD only a single person_assignment per approval (identified by role)
- 18 similar to R19<> only a single date_assignment per approval (identified by role)
- 19 R9 Version_history.current_version is mapped to the same place as version_history.versions[last].
- 20 R2
- 21 R7<extended_property_definition, 'versionable object', ID, ".*:">
- 22 R7<representation, 'versionable object', ID, ".*:">
- 23 R19<PRODUCT_DEFINITION_SHAPE, 'library manufacturing definition', 1, SHAPE_ASPECT, OF_SHAPE, NAME, 'library manufacturing definition parameters'>
- 24 R19<PRODUCT_DEFINITION_SHAPE, 'library definition', 1, SHAPE_ASPECT, OF_SHAPE, NAME, 'library definition parameters'>
- 25 R22<PRODUCT_DEFINITION_SHAPE, 'library manufacturing definition', 1, SHAPE_ASPECT, OF_SHAPE, 'source'>
- 26 R7<extended_product_definition_shape, 'versionable object', ID, ".*:">

27 R13<representation, 'library definition', items, 1>
 28 R26
 29 R19<PRODUCT_DEFINITION_SHAPE, 'manufacturing definition', 1, SHAPE_ASPECT,
 OF_SHAPE, NAME, 'manufacturing definition parameters'>
 30 R22<PRODUCT_DEFINITION_SHAPE, 'library definition', 1, SHAPE_ASPECT, OF_SHAPE,
 'source'>
 31 R11<'source', 'library definition'>
 32 R19<PRODUCT_DEFINITION_SHAPE, 'library design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, NAME, 'library design parameters'>
 33 R13<representation, 'library design definition', items, 1>
 34 R22<PRODUCT_DEFINITION_SHAPE, 'library design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, NAME, 'source'>
 35 R11<'source', 'library design definition'>
 36 R3
 37 R10<'design still water bending moment parameters', ['hogging amidship', 'sagging amidship']>
 38 R10<'design vertical wave bending moment parameters', ['hogging', 'sagging']>
 39 R10<'design vertical wave shear force parameters', ['negative value', 'positive value']>
 40 R10X<'table of shear force value', 'design SWSF values'>
 41 R10X<'maximum permissible still water bending moment',
 'maximum permissible SWBM values'>
 42 R10X<'maximum permissible still water shear force', 'maximum permissible SWSF values'>
 43 R19<PRODUCT_DEFINITION, 'design still water bending moment',
 1, PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'design still water bending moment parameters']>
 44 R19<PRODUCT_DEFINITION,
 'design still water shear force', 1, PROPERTY_DEFINITION_REPRESENTATION, DEFINITION,
 NAME, 'UNUSED. ']>
 45 R19<PRODUCT_DEFINITION,
 'design vertical wave bending moment', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'design vertical wave bending moment parameters']>
 46 R19<PRODUCT_DEFINITION,
 'design vertical wave shear force', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'design vertical wave shear force parameters']>
 47 R19<PRODUCT_DEFINITION,
 'maximum permissible still water bending moment', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'UNUSED. ']>
 48 R19<PRODUCT_DEFINITION,
 'maximum permissible still water shear force', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'UNUSED. ']>
 49 R11X<'design SWSF values', ['negative value', 'positive value', 'position']>
 50 R11X<'maximum permissible SWBM values', ['hogging', 'sagging', 'position']>
 51 R11X<'maximum permissible SWSF values', ['negative value', 'positive value', 'position']>
 52 The *external_source*, that is the image of an *universal_resource_locator* shall be instantiated in
 accordance to the rules of the W3C.
 53 All attributes of *universal_resource_locator* shall be mapped to the same instance of *external_source*,
 which is produced by concatenation of individual STRINGS.
 54 The attribute *source* of the entity type *external_identification_assignment* references the
external_source that specifies the context in which the value of the attribute *assigned_id* is defined.
 This instance of *external_source* is the same instance that is referenced from the
external_class_instance.
 55 The value of *externally_defined_class.item_id* (inherited from *externally_defined_item*) shall be
 instantiated in accordance to the class code of ISO 13584-42.
 56 The value of *library_class_version_assignment.assigned_id* (inherited from *identification_assignment*)
 shall be instantiated in accordance to the version code of ISO 13584-42.
 57 The value of *organisation.id* shall be instantiated in accordance to the supplier code of ISO 13584-26.
 58 R14<['internal location', 'street number', 'street', 'postal box', 'town', 'region', 'postal code',
 'country', 'facsimile number', 'telephone number', 'electronic mail address', 'telex number']>
 59 R16<['internal location', 'street number', 'street', 'postal box', 'town', 'region', 'postal code',
 'country', 'facsimile number', 'telephone number', 'electronic mail address', 'telex number']>

- 60 R17<['line_number', 'page', 'paragraph', 'section']>
 61 This id must be identical to the GUID of the placeholder for the external item. This placeholder is, e.g.
 a product definition for moulded form.
 62 At least one address component shall exist.
 63 The AIM specific subtype known_source shall have a local rule SELF\predefined_item.name =
 'ISO 13584 library'.
 64 Each instance of type known_source needs to be referenced by exactly one
 applied_organisation_assignment that plays the role 'library supplier'
 65 The AIM specific subtype library_class_version_assignment contains the local rule
 SELF\identification_assignment.role.name = 'class version'.
 66 The AIM specific subtype library_class_version_assignment contains the local rule
 SELF\external_identification_assignment.source refers to an instance that has the partial type
 known_source.
 67 The list of items in the AIM specific subtype library_class_version_assignment shall contain exactly
 one instance.
 68 For each externally_defined_general_property there exists (exactly?) one
 externally_defined_item_relationship relating it to an externally_defined_class. The name attribute of
 this externally_defined_item_relationship is equal to 'name scope'.
 69 The list of items in the AIM specific subtype library_property_version_assignment shall contain exactly
 one instance.
 70 The AIM specific subtype library_property_version_assignment has a local rule saying
 SELF\identification_assignment.role.name = 'property version'.
 71 The AIM specific subtype library_property_version_assignment has a local rule saying
 SELF\external_identification_assignment.source references an instance that has the partial type
 known_source.
 72 Cardinality: There should be 1 to many images of property_value associated to a certain part via a
 library_element_reference.
 73 The attribute items[i] of the AIM specific subtype applied_classification_assignment shall only be
 images of Library_definition or profile_cross_section if
 applied_classification_assignment.assigned_classification has a partial subtype
 externally_defined_class.
 74 If an applied_external_identification_assignment with role = 'external_reference' has an item of type
 applied_document_reference with role='document reference' then this must be the only item in the
 list.
 75 RJ32<'hull cross section', ['structural system', 'compartment']>: The group that is linked to a
 product_definition of class 'hull cross section' shall only collect group_items of type product_definition
 with class being either 'structural system' or 'compartment'.
 76 R21<product_definition, 'hull cross section', 1, extended_property_definition, definition, 'local co
 ordinate system with station reference'>: A product_definition of class 'hull cross section' shall be
 referenced by exactly one extended_property_definition of class 'local co ordinate system with station
 reference'.
 77 R57<product_definition, 'hull cross section', 1, 'context of revision', 'revision with context'>: A
 product_definition of class 'hull cross section' shall be referenced by at least one group of class 'revision
 with context' as 'context of revision'.
 78 RJ33<'hull cross section', 'structural system'>: A product_definition of class 'hull cross section' shall
 through its associated group reference at least one product_definition of class 'structural system'.
 79 R58<extended_product_definition_shape, 'hull cross section design definition', 1, 'members', 'revision
 with context'>: An extended_product_definition_shape of class 'hull cross section design definition'
 shall be referenced by at least one group of class 'revision with context' in its set of 'members'.
 80 R59<extended_product_definition_shape, 'hull cross section design definition', 1, 'members', 'revision
 with context'>: A group of class 'revision with context' shall have at maximum 1 group_assignment with
 role 'members' that references an extended_product_definition_shape of class 'hull cross section design
 definition'.
 81 R2: the attribute 'id' of the product_definition that is the image of a Hull_cross_section shall be globally
 unique.
 82 R60<Extended_product_definition_shape, 'hull cross section design definition', definition,
 product_definition, 'hull cross section', 'versions', 'version history'>:
 Extended_product_definition_shapes of class 'hull cross section design definition' that are 'defined for'

- the same product_definition of class 'hull cross section' shall belong to the same group of class 'version history' in its role as 'versions'.
- 83 R26: ensures that a type that is referenced by an applied_identification_assignment whose role is
 84 'version id' has exactly one reference of this type.
- 85 R2: Forces the global uniqueness of the attribute 'id' of the extended_product_definition_shape that is
 the image of a Hull_cross_section_design_definition.
- 86 R10<'hull cross section design definition parameters', ['mirrored symmetry']>: the 'items' attribute of the
 representation that is linked to an extended_product_definition_shape via a
 product_definition_representation that has the name "hull cross section design definition parameters"
 shall have exactly one representation_item with the name "mirrored symmetry".
- 87 R54<'hull cross section design definition parameters', 1>: the 'items' attribute of the representation that
 is linked to an extended_product_definition_shape via a product_definition_representation that has the
 name "hull cross section design definition parameters" shall have not more than one
 representation_item.
- 88 R19<shape_aspect, 'flare area', 2, property_definition_representation, definition, name, 'flare area
 parameters'>: A shape_aspect that is of class "flare area" shall be referenced by exactly two
 property_definition_representations that have the names "flare area parameters".
- 89 R3: the representation that is the image of Flare_area shall have a global_unit_assigned_context.
- 90 R10<'flare area parameters', ['area', 'height']>: the 'items' attribute of the representation that is linked to a
 shape_aspect via a product_definition_representation that has the name "flare area parameters" shall
 have exactly one representation_item with the name "area" and one with the name "height".
- 91 R54<'flare area parameters', 2>: the 'items' attribute of the representation that is linked to a
 shape_aspect via a product_definition_representation that has the name "flare area parameters" shall
 have not more than two representation_items.
- 92 R2
- 93 R7<PRODUCT, 'definable object', ID, ".*:">
- 94 WR1 of Global_id
- 95 R26
- 96 R40<'global axis placement'>
- 97 R10<'global axis placement', ['global axes and origin', 'after perpendicular offset', 'orientation']>
- 98 R3
- 99 R26
- 100 R2
- 101 R7<product_definition, 'global axis placement', id, ".*:">
- 102 R19<PROPERTY_DEFINITION, 'local co-ordinate system', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'local co-ordinate
 system'>
- 103 R10<'local co-ordinate system', ['local axes and origin']>
- 104 R19<PROPERTY_DEFINITION, 'local co-ordinate system', 1,
 TEM_DEFINED_TRANSFORMATION, TRANSFORM_ITEM_1, NAME, 'transformation to
 parent'>
- 105 R7<property_definition, 'local co-ordinate system', id, ".*:">
- 106 R7<property_definition, 'spacing table', id, ".*:">
- 107 R19<PROPERTY_DEFINITION, 'local co-ordinate system with position reference', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'local co-ordinate
 system with position reference'>
- 108 R10<'local co-ordinate system with position reference', ['local axes and origin']>
- 109 R44<'spacing position', ['position number', 'position']>
- 110 R44<'spacing position with offset', ['offset']>
- 111 R45<'spacing position with offset', 'spacing position'>
- 112 R2
- 113 R7<product_definition, 'definable object', ID, ".*:">
- 114 R7<product_definition_relationship, 'definable object', ID, ".*:">
- 115 R26
- 116 R7<extended_shape_aspect, 'definable object', ID, ".*:">
- 117 R19<SHAPE_ASPECT, 'composite feature', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'composite feature design parameters'>
- R7<extended_shape_aspect_relationship, 'definable object', ID, ".*:">

118 R3
 119 R7<representation, 'versionable object', ID, ".*:">
 120 R13<representation, 'explicit feature design definition', items, N>
 121 R11<'representations', 'explicit feature design definition'>
 122 R10<'composite feature design parameters', ['composed of', 'description', 'name']
 123 R7<extended_product_definition_shape, 'versionable object', ID, ".*:">
 124 R8<PRODUCT_DEFINITION, 'assembly relationship'>
 125 R40<'principal characteristics'>
 126 R10<'principal characteristics', ['length between perpendiculars', 'moulded breadth', 'moulded depth',
 'design draught', 'design deadweight', 'min draught at fp', 'max draught at fp', 'min draught at ap', 'max
 draught at ap']>
 127 R12<'principal characteristics', ['block coefficient']>
 128 R3
 129 R26
 130 R2
 131 R7<product_definition, 'principal characteristics', id, ".*:">
 132 R7<product_definition, 'class and statutory designation', id, ".*:">
 133 R7<product_definition, 'ship designation', id, ".*:">
 134 R7<product_definition, 'shipyard designation', id, ".*:">
 135 R7<product_definition, 'owner designation', id, ".*:">
 136 R7<product_related_product_category, 'ship type', id, ".*:">
 137 R40<'class and statutory designation'>
 138 R10<'class and statutory designation', ['class number']>
 139 R21<PRODUCT_DEFINITION, 'class and statutory designation', 1, PROPERTY_DEFINITION,
 DEFINITION, 'class notation'>
 140 R21<PRODUCT_DEFINITION, 'class and statutory designation', 1, PROPERTY_DEFINITION,
 DEFINITION, 'regulation'>
 141 R19<PROPERTY_DEFINITION, 'class notation', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'class notation'>
 142 R19<PROPERTY_DEFINITION, 'class notation', 1,
 APPLIED_ORGANIZATION_ASSIGNMENT, ITEMS, ROLE, 'class society'>
 143 R41<'class notation', ['class notations hull', 'class notations machinery']>
 144 R12<'class notation', ['ice class notation', 'service factor', 'approval required for heavy cargo']>
 145 R10<'class notation', ['service area', 'approval required for oil cargo', 'approval required for loading
 unloading aground', 'approval required for unloading grabs']>
 146 R40<'class parameters'>
 147 R10<'class parameters', ['length class', 'length solas', 'scantlings draught', 'block coefficient class',
 'design speed ahead', 'design speed astern']>
 148 R42<PRODUCT_DEFINITION, 'ship designation', 1,
 APPLIED_IDENTIFICATION_ASSIGNMENT, ITEMS, ROLE, NAME, 'IMO number', 'pennant
 hull number'>
 149 R19<PRODUCT_DEFINITION, 'ship designation', 1,
 APPLIED_IDENTIFICATION_ASSIGNMENT, ITEMS, ROLE, 'call sign'>
 150 R19<PRODUCT_DEFINITION, 'ship designation', 1,
 APPLIED_IDENTIFICATION_ASSIGNMENT, ITEMS, ROLE, 'flag state'>
 151 R19<PRODUCT_DEFINITION, 'ship designation', 1,
 APPLIED_IDENTIFICATION_ASSIGNMENT, ITEMS, ROLE, 'port of registration'>
 152 R19<PRODUCT_DEFINITION, 'shipyard designation', 1,
 APPLIED_ORGANIZATION_ASSIGNMENT, ITEMS, ROLE, 'shipyard'>
 153 R19<PRODUCT_DEFINITION, 'owner designation', 1,
 APPLIED_ORGANIZATION_ASSIGNMENT, ITEMS, ROLE, 'managing company'>
 154 R19<PRODUCT_DEFINITION, 'owner designation', 1,
 APPLIED_ORGANIZATION_ASSIGNMENT, ITEMS, ROLE, 'ordering company'>
 155 R19<PRODUCT_DEFINITION, 'owner designation', 1,
 APPLIED_ORGANIZATION_ASSIGNMENT, ITEMS, ROLE, 'owning company'>
 156 R40<'freeboard characteristics'>

157 R19<PRODUCT_DEFINITION, 'freeboard characteristics', 1,
 APPLIED_DATE_AND_TIME_ASSIGNMENT, ITEMS, ROLE, 'date freeboard assigned'>
 158 R19<PRODUCT_DEFINITION, 'freeboard characteristics', 1,
 APPLIED_ORGANIZATION_ASSIGNMENT, ITEMS, ROLE, 'freeboard assigned by'>
 159 R10<'freeboard characteristics', ['freeboard']>
 160 R21<PRODUCT_DEFINITION, 'freeboard characteristics', 1, PROPERTY_DEFINITION,
 DEFINITION, 'loadline'>
 161 R19<PROPERTY_DEFINITION, 'loadline', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'loadline'>
 162 R10<'loadline', ['load line length', 'load line depth', 'load line displacement', 'load line block
 coefficient', 'load line regulation']>
 163 R40<'lightship_definition'>
 164 R10<'lightship_definition', ['lightship weight', 'lightship centre of gravity']>
 165 R19<PROPERTY_DEFINITION, 'lightship weight item', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'lightship weight item'>
 166 R10<'lightship weight item', ['aft weight extent', 'fwd weight extent']>
 167 R19<PROPERTY_DEFINITION, 'weight and centre of gravity', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'weight and centre of
 gravity'>
 168 R10<'weight and centre of gravity', ['mass', 'centre of gravity']>
 169 R19<PROPERTY_DEFINITION_REPRESENTATION, 'weight and centre of gravity', 1,
 REPRESENTATION, USED_REPRESENTATION, NAME, 'moment 3d'>
 170 R43<'moment 3d', ['longitudinal moment', 'transverse moment', 'vertical moment', 'origin']>
 171 R26
 172 R2
 173 R3
 174 R7< EXTENDED_PRODUCT_DEFINITION_SHAPE, 'definition', ID, “*:”>
 175 R7< EXTENDED_SHAPE_ASPECT_RELATIONSHIP, 'definition', ID, “*:”>
 176 R10<'assembly manufacturing definition parameters', ['assembly method', 'assembly stage', 'mass', 'center
 of gravity']>
 177 R19<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'assembly manufacturing definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'assembly manufacturing
 definition parameters'>
 178 R10<'assembly manufacturing position parameters', ['assembly footprint', 'orientation']>
 179 R19<SHAPE_ASPECT, 'assembly manufacturing position', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'assembly
 manufacturing position parameters'>
 180 R19<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'plate manufacturing definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'outer contour
 representation'>
 181 R22<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'plate manufacturing definition', 1,
 SHAPE_ASPECT, OF_SHAPE, 'outer contour'>
 182 R10<'structural added material feature design definition parameter
 representation', ['added_material_length']>
 183 R12<'structural weld shrinkage allowance feature design definition parameter representation',
 ['longitudinal allowance', 'transverse allowance', 'vertical allowance']>
 184 R21<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'weld manufacturing definition', 1,
 SHAPE_ASPECT, OF_SHAPE, 'torch vector'>
 185 R10<'weld manufacturing definition parameters', ['position', 'process', 'degree of automations', 'number of
 weld passes', 'welding deposition sequences', 'welding environment', 'tack weld used']>
 186 R21<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'assembly manufacturing definition', 1,
 SHAPE_ASPECT, OF_SHAPE, 'centre of gravity'>
 187 R10<'welding sequence representation', ['name', 'start offset', 'end offset', 'direction']>
 188 R7<EXTENDED_PROPERTY_DEFINITION, ID, “*:”>
 189 R26
 190 R3
 191 R2

192 R19<PROPERTY_DEFINITION, 'ship material property', 1,
PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
'ship material property parameters'>

193 R10<'ship material property parameters', ['density']>

194 R19<PROPERTY_DEFINITION, 'ship material property', 1, APPLIED_DOCUMENT_REFERENCE,
ITEMS, ROLE.NAME, 'material reference'>

195 R19<PROPERTY_DEFINITION, 'homogeneous ship material property', 1,
PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'homogeneous ship
material property parameters'>

196 R10<'homogeneous ship material property parameters', ['density', 'poisson ratio', 'stress of fracture',
'thermal expansion coefficient', 'yield point', 'youngs module']>

197 R24<PROPERTY_DEFINITION, 'homogeneous ship material property', 1,
APPLIED_DOCUMENT_REFERENCE, ITEMS, ROLE.NAME, 'material reference'>

198 R55<derived_unit, name, 'angular stiffness unit', 4, elements>: The derived_unit with
derived_unit.name = 'angular stiffness unit' shall have maximum 4 derived_unit_elements.

199 R55<derived_unit, name, 'density unit', 2, elements >: The derived_unit with derived_unit.name =
'density unit' shall have maximum 2 derived_unit_elements.

200 R55<derived_unit, name, 'dilatation unit', 2, elements >: The derived_unit with derived_unit.name =
'dilatation unit' shall have maximum 2 derived_unit_elements.

201 R55<derived_unit, name, 'flow rate unit', 2, elements >: The derived_unit with derived_unit.name =
'flow rate unit' shall have maximum 2 derived_unit_elements.

202 R55<derived_unit, name, 'force unit', 3, elements >: The derived_unit with derived_unit.name = 'force
unit' shall have maximum 3 derived_unit_elements.

203 R55<derived_unit, name, 'frequency unit', 1, elements >: The derived_unit with derived_unit.name =
'frequency unit' shall have maximum 1 derived_unit_elements.

204 R55<derived_unit, name, 'heat rate unit', 3, elements >: The derived_unit with derived_unit.name =
'heat rate unit' shall have maximum 3 derived_unit_elements.

205 R55<derived_unit, name, 'inertia moment unit', 1, elements >: The derived_unit with derived_unit.name =
'inertia moment unit' shall have maximum 1 derived_unit_elements.

206 R55<derived_unit, name, 'linear stiffness unit', 3, elements >: The derived_unit with derived_unit.name =
'linear stiffness unit' shall have maximum 3 derived_unit_elements.

207 R55<derived_unit, name, 'pressure unit', 3, elements >: The derived_unit with derived_unit.name =
'pressure unit' shall have maximum 3 derived_unit_elements.

208 R55<derived_unit, name, 'product of inertia unit', 1, elements >: The derived_unit with
derived_unit.name = 'product of inertia unit' shall have maximum 1 derived_unit_elements.

209 R55<derived_unit, name, 'section modulus unit', 1, elements >: The derived_unit with
derived_unit.name = 'section modulus unit' shall have maximum 1 derived_unit_elements.

210 R55<derived_unit, name, 'speed unit', 2, elements >: The derived_unit with derived_unit.name = 'speed
unit' shall have maximum 2 derived_unit_elements.

211 R55<derived_unit, name, 'stress unit', 3, elements >: The derived_unit with derived_unit.name = 'stress
unit' shall have maximum 3 derived_unit_elements.

212 R55<derived_unit, name, 'voltage unit', 4, elements >: The derived_unit with derived_unit.name =
'voltage unit' shall have maximum 4 derived_unit_elements.

213 R1<uncertainty_measure_with_unit, description>: The uncertainty_measure_with_unit.description shall
be mandatory, not optional as it is in the IRs.

214 A class shall never participate in a group_assignment.

215 RJ31<'edge cutout functional definition'>: Representations of class 'edge cutout functional definition'
shall not have a representation_context of type global_unit_assigned_context.

216 R12<'seam curve relationship parameter', 'displacement'>: the representation that points at a shape-
_aspect_relationship of class "seam curve relationship" shall contain zero or one representation_item
with the name 'displacement'.

217 RJ29<'seam curve', 'seam curve relationship'>: A seam_curve_relationship may, if its seam_curve is a
curve and not a topological relationship, include zero or one representation_items.

218 R18<'seam design definition', 2, 'border', 'seam'>: there shall be maximum 2 Seams
(extended_shape_aspect of class seam) as border-attribute to Seam_design_definition (representation).

219 R13<representation, 'seam curve relationship', items, 0>: forces the set of items of a representation of
class seam_curve_relationship to have zero elements.

- 220 RJ30<property_definition , name, 'parent', 1, shape_aspect, 'structural feature', shape_of>: A
shape_aspect of class 'structural feature' shall reference exactly one roperty_definition with name
'parent'.
- 221 R26: ensures that there is only one identification_assignment with name "version_id" to point to an
extended_shape_aspect_relationship of class .
- 222 R3: ensures that the representation has a global_unit_assigned_context
- 223 R13<representation, 'bevel design definition', items, 3>: forces the items of representation of class ... to
have maximum ... elements
- 224 R11<'offset', 'bevel design definition'>: forces the items of representation of class ... to have exactly one
representation_item with name
- 225 R11<'x y angle', 'bevel design definition'>: forces the items of representation of class ... to have exactly
one representation_item with name
- 226 R13<representation, 'circular cutout design definition', items, 1>: forces the items of representation of
class ... to have maximum ... elements
- 227 R11<'radius', 'circular cutout design definition'>: forces the items of representation of class ... to have
exactly one representation_item with name
- 228 R13<representation, 'drain hole cutout design definition', items, 5>: forces the items of representation of
class ... to have maximum ... elements
- 229 R11<'depth', 'drain hole cutout design definition'>: forces the items of representation of class ... to have
exactly one representation_item with name
- 230 R11<'drain hole radius', 'drain hole cutout design definition'>: forces the items of representation of class
... to have exactly one representation_item with name
- 231 R11<'gap', 'drain hole cutout design definition'>: forces the items of representation of class ... to have
exactly one representation_item with name
- 232 R11<'gap radius', 'drain hole cutout design definition'>: forces the items of representation of class ... to
have exactly one representation_item with name
- 233 R11<'width', 'drain hole cutout design definition'>: forces the items of representation of class ... to have
exactly one representation_item with name
- 234 R13<representation, 'edge cutout functional definition', items, 1>: forces the items of representation of
class ... to have maximum ... elements
- 235 R11<'the function', 'edge cutout functional definition'>: forces the items of representation of class ... to
have exactly one representation_item with name
- 236 R13<representation, 'elliptical cutout design definition', items, 2>: forces the items of representation of
class ... to have maximum ... elements
- 237 R11<'half axis a', 'elliptical cutout design definition'>: forces the items of representation of class ... to
have exactly one representation_item with name
- 238 R11<'half axis b', 'elliptical cutout design definition'>: forces the items of representation of class ... to
have exactly one representation_item with name
- 239 R13<representation, 'elongated oval cutout design definition', items, 6>: forces the items of
representation of class ... to have maximum ... elements
- 240 R11<'distance', 'elongated oval cutout design definition'>: forces the items of representation of class ...
to have exactly one representation_item with name
- 241 R11<'r1', 'elongated oval cutout design definition'>: forces the items of representation of class ... to have
exactly one representation_item with name
- 242 R11<'r2', 'elongated oval cutout design definition'>: forces the items of representation of class ... to have
exactly one representation_item with name
- 243 R13<representation, 'free form interior cutout design definition', items, 1>: forces the items of
representation of class ... to have maximum ... elements
- 244 R11<'bounding curve', 'free form interior cutout design definition'>: forces the items of representation
of class ... to have exactly one representation_item with name
- 245 R13<representation, 'interior cutout functional definition', items, 1>: forces the items of representation
of class ... to have maximum ... elements
- 246 R11<'the function', 'interior cutout functional definition'>: forces the items of representation of class ...
to have exactly one representation_item with name
- 247 R13<representation, 'inward round corner design definition', items, 3>: forces the items of
representation of class ... to have maximum ... elements
- 248 R11<'x offset', 'inward round corner design definition'>: forces the items of representation of class ... to
have exactly one representation_item with name

- 249 R11<'y offset', 'inward round corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 250 R11<'radius', 'inward round corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 251 R13<'representation', 'outward round corner design definition', items, 3>: forces the items of representation of class ... to have maximum ... elements
- 252 R11<'x offset', 'outward round corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 253 R11<'y offset', 'outward round corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 254 R11<'radius', 'outward round corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 255 R13<'representation', 'part edge cutout design definition', items, 8>: forces the items of representation of class ... to have maximum ... elements
- 256 R11<'R left', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 257 R11<'R right', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 258 R11<'x A', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 259 R11<'x B', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 260 R11<'x C', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 261 R11<'x D', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 262 R11<'y B', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 263 R11<'y C', 'part edge cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 264 R13<'representation', 'rectangular cutback corner design definition', items, 4>: forces the items of representation of class ... to have maximum ... elements
- 265 R11<'x depth', 'rectangular cutback corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 266 R11<'y depth', 'rectangular cutback corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 267 R11<'radius', 'rectangular cutback corner design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 268 R13<'representation', 'rectangular cutout design definition', items, 2>: forces the items of representation of class ... to have maximum ... elements
- 269 R11<'length of', 'rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 270 R11<'width', 'rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 271 R13<'representation', 'round corner rectangular cutout design definition', items, 3>: forces the items of representation of class ... to have maximum ... elements
- 272 R11<'corner radius', 'round corner rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 273 R11<'length of', 'round corner rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 274 R11<'width', 'round corner rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 275 R13<'representation', 'round edge rectangular cutout design definition', items, 4>: forces the items of representation of class ... to have maximum ... elements
- 276 R11<'edge radius', 'round edge rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name
- 277 R11<'distance', 'round edge rectangular cutout design definition'>: forces the items of representation of class ... to have exactly one representation_item with name

- 278 R11<'length of', 'round edge rectangular cutout design definition': forces the items of representation of
class ... to have exactly one representation_item with name
- 279 R11<'width', 'round edge rectangular cutout design definition': forces the items of representation of
class ... to have exactly one representation_item with name
- 280 R28<'shape_aspect', 'seam curve relationship', 'related', 'seam', 'border': either the related_shape_aspect
of an extended_shape_aspect_relationship of class 'seam curve relationship' is an
extended_shape_aspect of class 'seam' or of class 'border'.
- 281 R27: the two shape_aspects that are related by a shape_aspect_relationship of class 'seam curve
relationship' shall not be identical.
- 282 R13<'representation', 'shear bevel design definition', items, 4>: forces the items of representation of class
... to have maximum ... elements
- 283 R11<'x z angle', 'shear bevel design definition': forces the items of representation of class ... to have
exactly one representation_item with name
- 284 R11<'offset', 'shear bevel design definition': forces the items of representation of class ... to have
exactly one representation_item with name
- 285 R11<'x y angle', 'shear bevel design definition': forces the items of representation of class ... to have
exactly one representation_item with name
- 286 R13<'representation', 'triangular cutout design definition', items, 6>: forces the items of representation of
class ... to have maximum ... elements
- 287 R11<'c2 x', 'triangular cutout design definition': forces the items of representation of class ... to have
exactly one representation_item with name
- 288 R11<'c3 x', 'triangular cutout design definition': forces the items of representation of class ... to have
exactly one representation_item with name
- 289 R11<'c3 y', 'triangular cutout design definition': forces the items of representation of class ... to have
exactly one representation_item with name
- 290 R11<'c1 radius', 'triangular cutout design definition': forces the items of representation of class ... to
have exactly one representation_item with name
- 291 R11<'c2 radius', 'triangular cutout design definition': forces the items of representation of class ... to
have exactly one representation_item with name
- 292 R11<'c3 radius', 'triangular cutout design definition': forces the items of representation of class ... to
have exactly one representation_item with name
- 293 R2
- 294 R3
- 295 R6<'function', 'user defined', 'user def function'>
- 296 R7<PRODUCT_DEFINITION, 'definable object', ID, '*', '*'>
- 297 R8<PRODUCT_DEFINITION, 'structural part symmetry relationship'>
- 298 R7<EXTENDED_PROPERTY_DEFINITION, 'definition', ID, '*', '*'>
- 299 R7<PRODUCT_DEFINITION_RELATIONSHIP, 'definable object', ID, '*', '*'>
- 300 R7<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'definition', ID, '*', '*'>
- 301 R10<'angle bar cross section design parameters', ['depth', 'width', 'radius', 'thk']>
- 302 R10<'bulb flat cross section design parameters', ['depth', 'bulb width', 'bulb radius', 'tw']>
- 303 R10<'channel cross section design parameters', ['depth', 'width', 'web thk', 'flange thk', 'radius', 'k']>
- 304 R10<'circular hollow profile cross section design parameters', ['outer diameter', 'inner diameter']>
- 305 R10<'corrugated part design parameters', ['mirrored', 'offset', 'repetition', 'thickness']>
- 306 R12<'corrugated part design parameters', ['mass']>
- 307 R10<'corrugation design parameters', ['depth', 'flat width 1', 'flat width 2', 'slope width']>
- 308 R12<'corrugation design parameters', ['radius 1', 'radius 2']>
- 309 R10<'explicit profile cross section shape', ['cross section geometry', 'local coordinate system']>
- 310 R10<'flanged plate cross section design parameters', ['depth', 'width', 'radius', 'thk']>
- 311 R10<'flat bar cross section design parameters', ['depth', 'width']>
- 312 R10<'plate design parameters', ['material offset', 'thickness']>
- 313 R12<'plate design parameters', ['mass']>
- 314 R12<'profile boundary relationship design parameters', ['displacement']>
- 315 R10<'profile curve trace line design parameters', ['curve']>
- 316 R12<'profile curve trace line design parameters', ['displacement']>
- 317 R10<'profile design parameters', ['mirrored']>
- 318 R12<'profile design parameters', ['mass']>
- 319 R12<'profile trace line relationship design parameters', ['displacement', 'auxiliary line']>
- 320 R10<'round bar cross section design parameters', ['diameter']>

321 R10<'section properties', ['nominal mass per length', 'area', 'na u', 'na v', 'moi u', 'moi v', 'moi uv', 'tr']>
 322 R10<'square tube cross section design parameters', ['depth', 'width', 'wall thk']>
 323 R10<'twist location design parameters', ['location', 'direction']>
 324 R10<'t bar cross section design parameters', ['depth', 'width', 'web thk', 'flange thk', 'radius', 'k']>
 325 R10<'w shape cross section', ['depth', 'width', 'web thk', 'flange thk', 'radius', 'k']>
 326 R21<PRODUCT_DEFINITION_SHAPE, 'corrugated part design definition', 2, SHAPE_ASPECT,
 OF_SHAPE, 'border'>
 327 R19<PRODUCT_DEFINITION_SHAPE, 'corrugated part design definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION,
 NAME, 'corrugated part design parameters'>
 328 R21<PRODUCT_DEFINITION_SHAPE, 'corrugated part design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, 'corrugation'>
 329 R15<SHAPE_ASPECT, 'corrugation', PROPERTY_DEFINITION_REPRESENTATION>
 330 R19<SHAPE_ASPECT, 'library profile cross section', 1,
 APPLIED_CLASSIFICATION_ASSIGNMENT, ITEMS, ROLE, 'definitional class membership'>
 331 R19<SHAPE_ASPECT, 'angle bar cross section', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION,
 NAME, 'angle bar cross section design parameters'>
 332 R19<SHAPE_ASPECT, 'bulbflat cross section', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'bulbflat cross section design parameters'>
 333 R19<SHAPE_ASPECT, 'channel cross section', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'channel cross section design parameters'>
 334 R19<SHAPE_ASPECT, 'circular hollow cross section', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'circular hollow cross
 section design parameters'>
 335 R19<SHAPE_ASPECT, 'explicit profile cross section', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'explicit profile cross
 section design parameters'>
 336 R19<SHAPE_ASPECT, 'flat bar cross section', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'flat bar cross section design parameters'>
 337 R19<SHAPE_ASPECT, 'flanged plate cross section', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'flanged plate cross section design parameters'>
 338 R22<PRODUCT_DEFINITION_SHAPE, 'plate design definition', 1, SHAPE_ASPECT, OF_SHAPE,
 'border'>
 339 R19<PRODUCT_DEFINITION_SHAPE, 'plate design definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'plate design parameters'>
 340 R23<PRODUCT_DEFINITION_SHAPE, 'plate design definition', 1, SHAPE_ASPECT, OF_SHAPE,
 'moulded surface'>
 341 R10<'plate function parameters', ['function']>
 342 R19<PROPERTY_DEFINITION, 'plate functional definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'plate function parameters'>
 343 R24<SHAPE_ASPECT, 'profile cross section', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'section properties'>
 344 R19<SHAPE_ASPECT, 'profile curve trace line', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'profile curve trace line design parameters'>
 345 R25<PRODUCT_DEFINITION_SHAPE, 'profile design definition', 0, 2, SHAPE_ASPECT,
 OF_SHAPE, 'border'>
 346 R19<PRODUCT_DEFINITION_SHAPE, 'profile design definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'profile design parameters'>
 347 R23<PRODUCT_DEFINITION_SHAPE, 'profile design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, 'profile curve trace line'>
 348 R21<PRODUCT_DEFINITION_SHAPE, 'profile design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, 'profile cross section'>
 349 R10<'profile function parameters', ['function']>

350 R19<PROPERTY_DEFINITION, 'profile functional definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'profile function parameters'>
 351 R19<SHAPE_ASPECT, 'round bar cross section', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'round bar cross section design parameters'>
 352 R12<'plate function parameters', ['user def function']>
 353 R12<'profile function parameters', ['user def function']>
 354 R10<'structural part symmetry relationship design parameters', ['mirroring plane']>
 355 R19<SHAPE_ASPECT, 'twist location', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'twist location design parameters'>
 356 R19<SHAPE_ASPECT, 't bar cross section', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 't bar cross section design parameters'>
 357 R19<SHAPE_ASPECT, 'w shape cross section', 1, PROPERTY_DEFINITION_REPRESENTATION,
 DEFINITION, NAME, 'w shape cross section design parameters'>
 358 R26
 359 R2
 360 R7<PRODUCT_DEFINITION, 'definable object', ID, “*. *“>
 361 R26
 362 R6<'function', 'user defined', 'user def function'>
 363 R6<'tightness', 'user defined', 'user defined tightness'>
 364 R7<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'definition', ID, “*. *“>
 365 R3
 366 R8<PRODUCT_DEFINITION, 'structural system symmetry relationship'>
 367 R12<'corrugated structure boundary relationship design parameters', ['displacement']>
 368 R22<PRODUCT_DEFINITION_SHAPE, 'corrugated structure design definition', 1,
 SHAPE_ASPECT, OF_SHAPE, 'border'>
 369 R19<PRODUCT_DEFINITION_SHAPE, 'corrugated structure design definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'corrugated structure design parameters'>
 370 R23<PRODUCT_DEFINITION_SHAPE, 'corrugated structure design definition', 1,
 SHAPE_ASPECT, OF_SHAPE, 'moulded surface'>
 371 R10<'corrugated structure design parameters', ['thickness', 'tightness']>
 372 R12<'corrugated structure design parameters', ['corrugation width', 'user defined tightness']>
 373 R19<PROPERTY_DEFINITION, 'corrugated structure functional definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'corrugated structure function parameters'>
 374 R12<'corrugated structure function parameters', ['user def function']>
 375 R10<'corrugated structure function parameters', ['function']>
 376 R12<'panel system boundary relationship design parameters', ['displacement']>
 377 R19<SHAPE_ASPECT, 'panel system curve boundary', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'panel system curve boundary design parameters'>
 378 R12<'panel system curve boundary design parameters', ['displacement']>
 379 R10<'panel system curve boundary design parameters', ['curve']>
 380 R10<'panel system design parameters', ['material offset', 'thickness', 'tightness']>
 381 R12<'panel system design parameters', ['stiffener spacing', 'user defined tightness']>
 382 R22<PRODUCT_DEFINITION_SHAPE, 'panel system design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, 'border'>
 383 R19<PRODUCT_DEFINITION_SHAPE, 'panel system design definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'panel system design parameters'>
 384 R23<PRODUCT_DEFINITION_SHAPE, 'panel system design definition', 1, SHAPE_ASPECT,
 OF_SHAPE, 'moulded surface'>
 385 R19<PROPERTY_DEFINITION, 'panel system functional definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'panel system function parameters'>
 386 R10<'panel system function parameters', ['function']>
 387 R12<'panel system function parameters', ['user def function']>

388 R19<SHAPE_ASPECT, 'panel system plane boundary', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'panel system plane boundary design parameters'>
 389 R10<'panel system plane boundary design parameters', ['plane']>
 390 R19<PROPERTY_DEFINITION, 'plate strake functional definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'plate strake function parameters'>
 391 R10<'plate strake function parameters', ['function']>
 392 R12<'plate strake function parameters', ['user def function']>
 393 R32<product_definition, 'class membership', 'user defined', 'user defined class'>
 394 R12<'structural system adjacency relationship design parameters', ['displacement']>
 395 R10<'structural system design parameters', ['tightness']>
 396 R12<'structural system design parameters', ['user def tightness']>
 397 R10<'structural system symmetry relationship design parameters', ['mirroring plane']>
 398 RJ33<'panel system', 'plate'>
 399 R7<PRODUCT_DEFINITION_RELATIONSHIP, 'definable object', ID, “*.:*“>
 400 R19<PRODUCT_DEFINITION_SHAPE, 'structural system design definition', 1,
 PROPERTY_DEFINITION_REPRESENTATION, DEFINITION, NAME,
 'structural system design parameters'>
 401 R7<EXTENDED_PROPERTY_DEFINITION, 'definition', ID, “*.:*“>
 402 R3
 403 R2
 404 R7<PRODUCT_DEFINITION, 'definable object', ID, “*.:*“>
 405 R26
 406 R19<PRODUCT_DEFINITION_SHAPE, 'beveled groove weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'beveled groove weld
 design parameters'>
 407 R7<PRODUCT_DEFINITION_RELATIONSHIP, 'definable object', ID, “*.:*“>
 408 R10<'beveled groove weld design parameters', ['endcut shape', 'taper', 'sideness', 'weld joint spacer',
 'configuration', 'penetration', 'joint_orientation', 'tightness']>
 409 R7<REPRESENTATION, 'definition', ID, “*.:*“>
 410 R19<PRODUCT_DEFINITION_SHAPE, 'butt groove weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'butt groove weld
 design parameters'>
 411 R7<EXTENDED_PRODUCT_DEFINITION_SHAPE, 'definition', ID, “*.:*“>
 412 R10<'butt groove weld design parameters', ['face shape', 'sideness', 'weld joint spacer',
 'configuration', 'penetration', 'joint_orientation', 'tightness']>
 413 R7<EXTENDED_PROPERTY_DEFINITION, 'definition', ID, “*.:*“>
 414 R19<PRODUCT_DEFINITION_SHAPE, 'fillet weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'fillet weld design
 parameters'>
 415 R19<PRODUCT_DEFINITION_SHAPE, 'continuous fillet weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'continuous fillet weld
 design parameters'>
 416 R10<'electrode chemical composition design parameters', ['carbon', 'silicon', 'manganese',
 'phosphorus', 'sulphur', 'nickeö', 'molydenum', 'aluminium', 'titanium', 'zirconium', 'chromium',
 'copper', 'vanadium']>
 417 R10<'fillet weld design parameters', ['endcut shape type', 'sideness', 'configuration', 'penetration',
 'joint_orientation', 'tightness']>
 418 R19<PRODUCT_DEFINITION_SHAPE, 'groove weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'groove weld design
 parameters'>
 419 R10<'groove weld design parameters', ['sideness', 'weld joint spacer', 'configuration', 'penetration',
 'joint_orientation', 'tightness']>
 420 R19<PRODUCT_DEFINITION_SHAPE, 'intermittent fillet weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'intermittent fillet weld
 design parameters'>

421 R10<'intermittent fillet weld design parameters', ['end rules', 'cutout rules', 'fillet alignment', 'fillet
 weld length', 'fillet weld spacing', 'penetration rules', 'endcut shape type', 'sideness', 'configuration',
 'penetration', 'joint_orientation', 'tightness']>
 422 R19<PRODUCT_DEFINITION_SHAPE, 'spot seam weld', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'spot seam weld design
 parameters'>
 423 R10<'spot seam weld design parameters', ['weld context', 'configuration', 'penetration',
 'joint_orientation', 'tightness']>
 424 R10<'weld design parameters', ['shape of weld surface', 'weld geometry', 'connection angle', 'rotation
 angle', 'inclination angle', 'shrinkage', 'weld size']>
 425 R19<PRODUCT_DEFINITION_SHAPE, 'weld filler material', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'weld filler material design
 parameters'>
 426 R12<'weld filler material design parameters', ['chemical composition id', 'electrode id', 'material
 strength id', 'notch impact work id']>
 427 R19<PRODUCT_DEFINITION_SHAPE, 'welded joint design definition', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'welded joint design
 parameters'>
 428 R10<'welded joint design parameters', ['configuration', 'penetration', 'joint_orientation', 'tightness']>
 429 R12<'weld design parameters', ['start suspension', 'end suspension', 'weld cross section area']>
 430 R12<'groove weld design parameters', ['weld backing type']>
 431 R12<'beveled groove weld design parameters', ['weld backing type', 'taper angle']>
 432 R19<PRODUCT_DEFINITION_SHAPE, 'structural part joint design definition', 1,
 PRODUCT_DEFINITION_REPRESENTATION, DEFINITION, NAME, 'structural part joint design
 parameters'>
 433 R12<'butt groove weld design parameters', ['weld backing type']>
 434 R10<'structural part joint design parameters', ['tightness', 'joint orientation']>
 435 R10<'continuous fillet weld design parameters', ['endcut shape type', 'sideness', 'configuration',
 'penetration', 'joint_orientation', 'tightness']>

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources (and the AICs) and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the text for constructs that are imported from the integrated resources (and the AICs). The definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes that are not imported into the AIM. Requirements stated in the integrated resources that refer to such items and subtypes apply exclusively to those items that are imported into the AIM.

EXPRESS specification:

```

*)
SCHEMA ship_structures_scheme;

    USE FROM aic_advanced_brep;                                -- ISO 10303-514

    USE FROM aic_edge_based_wireframe;                        -- ISO 10303-501

    USE FROM aic_geometrically_bounded_wireframe;            -- ISO 10303-510

    USE FROM aic_non_manifold_surface;                        -- ISO 10303-508

    USE FROM aic_topological_bounded_surface;                 -- ISO 10303-511

    USE FROM action_schema                                    -- ISO 10303-41
        (action,
         action_request_solution,
         executed_action,
         versioned_action_request);

    USE FROM approval_schema                                -- ISO 10303-41
        (approval,
         approval_status);

    USE FROM date_time_schema;                                -- ISO 10303-41
        (date_and_time);

    USE FROM document_schema                                  -- ISO 10303-41
        (document,
         document_usage_constraint);

    USE FROM external_reference_schema;                       -- ISO 10303-41
        (external_source,
         externally_defined_item);

    USE FROM geometry_schema                                  -- ISO 10303-42
        (axis2_placement_3d,

```

```

        bounded_curve,
        cartesian_point,
        curve,
        direction);

USE FROM group_schema                                -- ISO 10303-41
    (group);

USE FROM management_resources_schema                  -- ISO 10303-41
    (classification_assignment,
     document_reference,
     external_identification_assignment,
     group_assignment,
     identification_assignment,
     organization_assignment,
     person_and_organization_assignment,
     person_assignment);

USE FROM material_property_representation_schema      -- ISO 10303-45
    (property_definition_relationship);

USE FROM measure_schema                              -- ISO 10303-41
    (area_measure_with_unit,
     area_unit;
     conversion_based_unit,
     derived_unit,
     derived_unit_element,
     dimensional_exponents,
     global_unit_assigned_context,
     length_measure_with_unit,
     length_unit,
     luminous_intensity_measure_with_unit,
     luminous_intensity_unit,
     mass_measure_with_unit,
     mass_unit,
     measure_with_unit,
     named_unit,
     plane_angle_measure_with_unit,
     plane_angle_unit,
     positive_length_measure_with_unit,
     positive_length_unit,
     ratio_measure_with_unit,
     ratio_unit,
     si_unit,
     thermodynamic_temperature_measure_with_unit,
     thermodynamic_temperature_unit,
     time_measure,
     time_unit,
     uncertainty_measure_with_unit,
     volume_measure_with_unit,
     volume_unit);

```

```

USE FROM person_organization_schema                -- ISO 10303-41
  (organization,
   organizational_project);

USE FROM product_definition_schema                -- ISO 10303-41
  (product,
   product_category,
   product_definition,
   product_definition_relationship,
   product_related_product_category);

USE FROM product_property_definition_schema        -- ISO 10303-41
  (general_property,
   product_definition_shape,
   property_definition,
   property_definition_relationship,
   shape_aspect,
   shape_aspect_relationship);

USE FROM qualified_measure_schema                -- ISO 10303-45
  (descriptive_representation_item,
   material_designation);

USE FROM representation_schema                  -- ISO 10303-43
  (compound_representation_item,
   global_uncertainty_assigned_context,
   representation,
   representation_context,
   representation_item,
   representation_relationship,
   value_representation_item);

USE FROM support_resource_schema                -- ISO 10303-41
  (identifier);
END_SCHEMA;
(*)

```

NOTE The schemas referenced above can be found in the following parts of ISO 10303:

aic_advanced_brep	ISO 10303-514
aic_egde_based_wireframe	ISO 10303-501
aic_geometrically_bounded_wireframe	ISO 10303-510
aic_non_manifold_surface	ISO 10303-508
aic_topological_bounded_surface;	ISO 10303-511
action_schema	ISO 10303-41
approval_schema	ISO 10303-41
date_time_schema	ISO 10303-41
document_schema	ISO 10303-41

geometry_schema	ISO 10303-42
group_schema	ISO 10303-41
management_resources_schema	ISO 10303-41
measure_schema	ISO 10303-41
person_organization_schema	ISO 10303-41
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
support_resource_schema	ISO 10303-41

5.2.1 Fundamental concepts and assumptions

5.2.2 Ship structures type definitions

5.2.2.1 action_request_item

An action_request_item is an aspect of the ship structures, which has an action_request assigned.

EXPRESS specification:

```

TYPE action_request_item = SELECT (
    action,
    executed_action,
    group,
    product,
    product_definition,
    product_definition_relationship,
    product_definition_shape,
    property_definition);
END_TYPE;

```

5.2.2.2 approval_item

An approval_item is an aspect of the ship structures, which has an approval assigned.

EXPRESS specification:

```

TYPE approval_item = SELECT (
    extended_product_definition_shape,
    product_definition,
    product_definition_shape,
    property_definition);
END_TYPE;

```

5.2.2.3 classification_item

A classification_item is an aspect of the ship structures, to which a class is assigned.

EXPRESS specification:

```
TYPE classification_item = SELECT (
    extended_product_definition_shape,
    extended_property_definition,
    group,
    product,
    product_definition,
    product_definition_with_associated_document,
    product_definition_relationship,
    product_definition_shape,
    property_definition,
    property_definition_relationship,
    representation,
    shape_aspect);
END_TYPE;
```

5.2.2.4 document_reference_item

A document_reference_item is an aspect of the ship structures, to which a document is assigned.

EXPRESS specification:

```
TYPE document_reference_item = SELECT (
    group,
    property_definition);
END_TYPE;
```

5.2.2.5 external_identification_item

An external_identification_item is an aspect of the ship structures, which is externally identified.

EXPRESS specification:

```
TYPE external_identification_item = SELECT (
    document_reference,
    extended_product_definition_shape);
END_TYPE;
```

5.2.2.6 identification_item

An identification_item is an aspect of the ship structures, to which an identifier is assigned.

EXPRESS specification:

```
TYPE identification_item = SELECT (  
    extended_property_definition,  
    extended_product_definition_shape,  
    extended_shape_aspect,  
    extended_shape_aspect_relationship,  
    externally_defined_class,  
    externally_defined_general_property,  
    group,  
    product,  
    product_definition,  
    product_definition_relationship,  
    property_definition,  
    property_definition_relationship,  
    representation  
    shape_aspect,  
    shape_aspect_relationship);  
END_TYPE;
```

5.2.2.7 group_item

A group_item is an aspect of the ship structures, which is assigned to a group.

EXPRESS specification:

```
TYPE group_item = SELECT (  
    approval,  
    product_definition,  
    product_definition_shape,  
    product_definition_with_associated_documen,  
    product_definition_relationship,  
    representation);  
END_TYPE;
```

5.2.2.8 organization_item

An organization_item is an aspect of the ship structures, to which an organization is assigned.

EXPRESS specification:

```

TYPE organization_item = SELECT (
    product_definition,
    property_definition);
END_TYPE;

```

5.2.3 Ship structures entity definitions**5.2.3.1 applied_action_request_assignment**

An `applied_action_request_assignment` assigns an `action_request` for a set of `action_request_item` instances.

EXPRESS specification:

```

ENTITY applied_action_request_assignment
    SUBTYPE OF (action_request_assignment);
    items: SET[1:?] OF action_request_item;
END_ENTITY;

```

Attribute definitions:

items: the instances of `action_request_solution`, `excuted_action`, `extended_product_related_product_category`, `group`, `product`, `product_definition`, `product_definition_relationship`, `product_definition_shape` or `property_definition`, to which an action is assigned.

5.2.3.2 applied_approval_assignment

An `applied_approval_assignment` assigns an approval for a set of `approval_item` instances.

EXPRESS specification:

```

ENTITY applied_approval_assignment
    SUBTYPE OF (approval_assignment);
    items: SET[1:?] OF approval_item;
END_ENTITY;

```


Attribute definitions:

items: the instances of `extended_product_definition_shape`, `extended_product_related_product_category`, `product`, `product_definition`, `product_definition_shape` or `property_definition`, to which an approval is assigned.

5.2.3.3 applied_classification_assignment

An `applied_classification_assignment` assigns a class for a set of `classification_item` instances.

EXPRESS specification:

```
ENTITY applied_classification_assignment
  SUBTYPE OF (classification_assignment);
  items: SET[1:?] OF classification_item;
END_ENTITY;
```

Attribute definitions:

items: the instances of `extended_product_definition_shape`, `extended_property_definition`, `group`, `product`, `product_definition`, `product_definition_relationship`, `product_definition_with_associated_document`, `product_definition_shape`, `property_definition`, `property_definition_relationship`, `representation` or `shape_aspect`, to which a class is assigned.

Associated global rules:

- `PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_manufacturing_definition_parameters` (see 5.2.5.10)
- `PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_definition_parameters` (see 5.2.5.11)
- `PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE` (see 5.2.5.12)
- `PRODUCT_DEFINITION_SHAPE_w_class_id_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters` (see 5.2.5.16)
- `PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE` (see 5.2.5.17)

- PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE (see 5.2.5.19)
- PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE (see 5.2.5.21)
- PRODUCT_DEFINITION_w_class_id_design_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_still_water_bending_moment_parameters (see 5.2.5.29)
- PRODUCT_DEFINITION_w_class_id_design_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED. (see 5.2.5.30)
- PRODUCT_DEFINITION_w_class_id_design_vertical_wave_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_vertical_wave_bending_moment_parameters (see 5.2.5.31)
- PRODUCT_DEFINITION_w_class_id_design_vertical_wave_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_vertical_wave_shear_force_parameters (see 5.2.5.32)
- PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED. (see 5.2.5.33)
- PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED. (see 5.2.5.34)
- shape_aspect_w_class_id_flare_area_referenced_by_exactly_2_property_definition_representation_via_definition_w_name_eq_flare_area_parameters (see 5.2.5.42)
- product_definition_w_class_id_global_axis_placement_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.47)
- PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system (see 5.2.5.51)
- PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_exactly_1_ITEM_DEFINED_TRANSFORMATION_via_TRANSFORM_ITEM_1_w_NAME_eq_transformation_to_parent (see 5.2.5.53)
- PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_with_position_reference_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION

- INITIATION_w_NAME_eq_local_co_ordinate_system_with_position_reference (see 5.2.5.56)
- SHAPE_ASPECT_w_class_id_composite_feature_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_composite_feature_design_parameters (see 5.2.5.65)
- product_definition_w_class_id_principal_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.73)
- product_definition_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.83)
- PRODUCT_DEFINITION_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_class_notation_via_DEFINITION (see 5.2.5.85)
- PRODUCT_DEFINITION_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_regulation_via_DEFINITION (see 5.2.5.86)
- PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_class_notation (see 5.2.5.87)
- PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_class_society (see 5.2.5.88)
- product_definition_w_class_id_class_parameters_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.92)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE.NAME_eq_IMO_number_or_pennant_hull_number (see 5.2.5.94)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_call_sign (see 5.2.5.95)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_flag_state (see 5.2.5.96)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_port_of_registration (see 5.2.5.97)
- PRODUCT_DEFINITION_w_class_id_shipyard_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard (see 5.2.5.98)

- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company (see 5.2.5.99)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_ordering_company (see 5.2.5.100)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_owning_company (see 5.2.5.101)
- product_definition_w_class_id_freeboard_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.102)
- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_DATE_AND_TIME_ASSIGNMENT_via_ITEMS_w_ROLE_eq_date_freeboard_assigned (see 5.2.5.103)
- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_assigned_by (see 5.2.5.104)
- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_loadline_via_DEFINITION (see 5.2.5.106)
- PROPERTY_DEFINITION_w_class_id_loadline_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_loadline (see 5.2.5.107)
- product_definition_w_class_id_lightship_definition_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.109)
- PROPERTY_DEFINITION_w_class_id_lightship_weight_item_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_lightship_weight_item (see 5.2.5.111)
- PROPERTY_DEFINITION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weight_and_centre_of_gravity (see 5.2.5.113)
- PROPERTY_DEFINITION_REPRESENTATION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_REPRESENTATION_via_USED_REPRESENTATION_w_NAME_eq_moment_3d (see 5.2.5.115)
- PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq-

- ship_material_property_parameters (see 5.2.5.118)
- PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE_NAME_eq_material_reference (see 5.2.5.120)
- PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_homogeneous_ship_material_property_parameters (see 5.2.5.121)
- PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_at_most_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE_NAME_eq_material_reference (see 5.2.5.123)
- EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_assembly_manufacturing_definition_parameters (see 5.2.5.215)
- SHAPE_ASPECT_w_class_id_assembly_manufacturing_position_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_assembly_manufacturing_position_parameters (see 5.2.5.217)
- EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_outer_contour_representation (see 5.2.5.218)
- EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_outer_contour_via_OF_SHAPE (see 5.2.5.219)
- EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_weld_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_torch_vector_via_OF_SHAPE (see 5.2.5.222)
- EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_centre_of_gravity_via_OF_SHAPE (see 5.2.5.224)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_exactly_2_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.256)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_part_design_parameters (see 5.2.5.257)

- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_corrugation_via_OF_SHAPE (see 5.2.5.258)
- SHAPE_ASPECT_w_class_id_corrugation_referenced_exactly_once_by_PROPERTY_DEFINITION_REPRESENTATION (see 5.2.5.259)
- SHAPE_ASPECT_w_class_id_angle_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_angle_bar_cross_section_design_parameters (see 5.2.5.260)
- SHAPE_ASPECT_w_class_id_bulbflat_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_bulbflat_cross_section_design_parameters (see 5.2.5.261)
- SHAPE_ASPECT_w_class_id_channel_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_channel_cross_section_design_parameters (see 5.2.5.262)
- SHAPE_ASPECT_w_class_id_circular_hollow_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_circular_hollow_cross_section_design_parameters (see 5.2.5.263)
- SHAPE_ASPECT_w_class_id_explicit_profile_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_explicit_profile_cross_section_design_parameters (see 5.2.5.264)
- SHAPE_ASPECT_w_class_id_flat_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flat_bar_cross_section_design_parameters (see 5.2.5.265)
- SHAPE_ASPECT_w_class_id_flanged_plate_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flanged_plate_cross_section_design_parameters (see 5.2.5.266)
- PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.267)
- PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_design_parameters (see 5.2.5.268)
- PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE (see 5.2.5.269)
- PROPERTY_DEFINITION_w_class_id_plate_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq-

- plate_function_parameters (see 5.2.5.271)
- SHAPE_ASPECT_w_class_id_profile_cross_section_referenced_by_at_most_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_section_properties (see 5.2.5.272)
- SHAPE_ASPECT_w_class_id_profile_curve_trace_line_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_curve_trace_line_design_parameters (see 5.2.5.273)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_N_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.274)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_N_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.275)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_profile_curve_trace_line_via_OF_SHAPE (see 5.2.5.276)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_profile_cross_section_via_OF_SHAPE (see 5.2.5.277)
- PROPERTY_DEFINITION_w_class_id_profile_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_function_parameters (see 5.2.5.279)
- SHAPE_ASPECT_w_class_id_round_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_round_bar_cross_section_design_parameters (see 5.2.5.280)
- SHAPE_ASPECT_w_class_id_twist_location_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_twist_location_design_parameters (see 5.2.5.284)
- SHAPE_ASPECT_w_class_id_t_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_t_bar_cross_section_design_parameters (see 5.2.5.285)
- SHAPE_ASPECT_w_class_id_w_shape_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_w_shape_cross_section_design_parameters (see 5.2.5.286)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.292)

- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_structure_design_parameters (see 5.2.5.293)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_ (see 5.2.5.294)
- PROPERTY_DEFINITION_w_class_id_corrugated_structure_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_structure_function_parameters (see 5.2.5.297)
- SHAPE_ASPECT_w_class_id_panel_system_curve_boundary_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_curve_boundary_design_parameters (see 5.2.5.301)
- PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.306)
- PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.307)
- PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE (see 5.2.5.308)
- PROPERTY_DEFINITION_w_class_id_panel_system_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_function_parameters (see 5.2.5.309)
- SHAPE_ASPECT_w_class_id_panel_system_plane_boundary_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_plane_boundary_design_parameters (see 5.2.5.312)
- PROPERTY_DEFINITION_w_class_id_plate_strake_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_strake_function_parameters (see 5.2.5.314)
- PRODUCT_DEFINITION_SHAPE_w_class_id_structural_system_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_structural_system_design_parameters (see 5.2.5.323)
- PRODUCT_DEFINITION_SHAPE_w_class_id_beveled_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_beveled_groove_weld_design_parameters (see 5.2.5.325)
- PRODUCT_DEFINITION_SHAPE_w_class_id_butt_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_

- butt_groove_weld_design_parameters (see 5.2.5.328)
- PRODUCT_DEFINITION_SHAPE_w_class_id_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_fillet_weld_design_parameters (see 5.2.5.330)
- PRODUCT_DEFINITION_SHAPE_w_class_id_continuous_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_continuous_fillet_weld_design_parameters (see 5.2.5.331)
- PRODUCT_DEFINITION_SHAPE_w_class_id_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_groove_weld_design_parameters (see 5.2.5.334)
- PRODUCT_DEFINITION_SHAPE_w_class_id_intermittent_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_intermittent_fillet_weld_design_parameters (see 5.2.5.336)
- PRODUCT_DEFINITION_SHAPE_w_class_id_spot_seam_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_spot_seam_weld_design_parameters (see 5.2.5.338)
- PRODUCT_DEFINITION_SHAPE_w_class_id_weld_filler_material_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weld_filler_material_design_parameters (see 5.2.5.341)
- PRODUCT_DEFINITION_SHAPE_w_class_id_welded_joint_design_definition_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_welded_joint_design_parameters (see 5.2.5.343)
- PRODUCT_DEFINITION_SHAPE_w_class_id_structural_part_joint_design_definition_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_structural_part_joint_design_parameters (see 5.2.5.348)
- one_rep_item_in_compound_rep_item_if_class_eq_spacing_position (see 5.2.5.58))
- one_rep_item_in_compound_rep_item_if_class_eq_spacing_position_with_offset (see 5.2.5.59))
- one_compound_rep_item_with_class_spacing_position_in_compound_rep_item_with_class_spacing_position_with_offset (see 5.2.5.60))
- ID_of_extended_property_definition_with_class_versionable_object_pattern (see 5.2.5.8)
- ID_of_representation_with_class_versionable_object_pattern (see 5.2.5.9)
- ID_of_extended_product_definition_shape_with_class_versionable_object_pattern (see 5.2.5.13)

- ID_of_PRODUCT_with_class_definable_object_pattern (see 5.2.5.45)
- id_of_product_definition_with_class_global_axis_placement_pattern (see 5.2.5.50)
- id_of_property_definition_with_class_local_co_ordinate_system_pattern (see 5.2.5.54)
- id_of_property_definition_with_class_spacing_table_pattern (see 5.2.5.55)
- ID_of_product_definition_with_class_definable_object_pattern (see 5.2.5.61)
- ID_of_product_definition_relationship_with_class_definable_object_pattern (see 5.2.5.62)
- ID_of_extended_shape_aspect_with_class_definable_object_pattern (see 5.2.5.64)
- ID_of_extended_shape_aspect_relationship_with_class_definable_object_pattern (see 5.2.5.66)
- ID_of_representation_with_class_versionable_object_pattern (see 5.2.5.67)
- ID_of_extended_product_definition_shape_with_class_versionable_object_pattern (see 5.2.5.71)
- id_of_product_definition_with_class_principal_characteristics_pattern (see 5.2.5.77)
- id_of_product_definition_with_class_class_and_statutory_designation_pattern (see 5.2.5.78)
- id_of_product_definition_with_class_ship_designation_pattern (see 5.2.5.79)
- id_of_product_definition_with_class_shipyard_designation_pattern (see 5.2.5.80)
- id_of_product_definition_with_class_owner_designation_pattern (see 5.2.5.81)
- id_of_product_related_product_category_with_class_ship_type_pattern (see 5.2.5.82)
- ID_of_EXTENDED_PRODUCT_DEFINITION_SHAPE_with_class_definition_pattern
(see 5.2.5.212)
- ID_of_EXTENDED_SHAPE_ASPECT_RELATIONSHIP_with_class_definition_pattern
(see 5.2.5.213)
- ID_of_PRODUCT_DEFINITION_with_class_definable_object_pattern (see 5.2.5.227)
- ID_of_PRODUCT_DEFINITION_RELATIONSHIP_with_class_definable_object_pattern
(see 5.2.5.229)
- ID_of_EXTENDED_PROPERTY_DEFINITION_with_class_definition_pattern (see 5.2.5.230)
- ID_of_REPRESENTATION_with_class_definition_pattern (see 5.2.5.327)

5.2.3.4 applied_document_reference

An applied_document_reference provides a reference to a document for the document_reference_item instances.

EXPRESS specification:

```
ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
  items: SET[1:?] OF document_reference_item;
END_ENTITY;
```

Attribute definitions:

items: the instances of group or property_definition, to which a document is assigned.

5.2.3.5 applied_external_identification_assignment

An applied_external_identification_assignment assigns an identifier to a set of externally defined external_identification_item instances.

EXPRESS specification:

```
ENTITY applied_external_identification_assignment
  SUBTYPE OF (external_identification_assignment);
  items: SET[1:?] OF external_identification_item;
END_ENTITY;
```

Attribute definitions:

items: the instances of document_reference or extended_product_definition_shape, to which an external_identification is assigned.

5.2.3.6 applied_identification_assignment

An applied_identification_assignment assigns an identifier for a set of identification_item instances.

EXPRESS specification:

```

ENTITY applied_identification_assignment
  SUBTYPE OF (identification_assignmen);
  items: SET[1:?] OF identification_item;
END_ENTITY;

```

Attribute definitions:

items: the instances of extended_property_definition, extended_product_definition_shape, extended_product_related_product_category, extended_shape_aspect, extended_shape_aspect_relationship, externally_defined_class, externally_defined_general_property, group, product, product_definition, product_definition_relationship, property_definition, property_definition_relationship, representation_shape_aspect or shape_aspect_relationship, to which an identifier is assigned.

5.2.3.7 applied_group_assignment

The applied_group_assignment

EXPRESS specification:

```

ENTITY applied_group_assignment
  SUBTYPE OF (group_assignment);
  items: SET[1:?] OF group_item;
  WHERE
    WR1: -- if role = 'equivalence' than items shall only have one entry
END_ENTITY;

```

Attribute definitions:

items: the instances of approval, product_definition, product_definition_shape, product_definition_with_associated_document, product_definition_relationship or representation, which are assigned to a group.

5.2.3.8 applied_organization_assignment

An applied_organization_assignment assigns an organization to a set of organization_item instances.

EXPRESS specification:

```
ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
  items: SET[1:?] OF organization_item;
END_ENTITY;
```

Attribute definitions:

items: the instances of product_definition or property_definition, to which an organization is assigned.

Associated global rules:

- PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_class_society (see 5.2.5.88)
- PRODUCT_DEFINITION_w_class_id_shipyard_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard (see 5.2.5.98)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company (see 5.2.5.99)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_ordering_company (see 5.2.5.100)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_owning_company (see 5.2.5.101)
- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_assigned_by (see 5.2.5.104)

5.2.3.9 class

A class is a group that is used to classify any aspect of the ship structures.

EXPRESS specification:

```

ENTITY class
  SUBTYPE OF (group);
END_ENTITY;

```

5.2.3.10 extended_material_designation

An extended_material_designation is a material_designation with a derived id attribute.

EXPRESS specification:

```

ENTITY extended_material_designation
  SUBTYPE OF (material_designation);
  DERIVED
    id: identifier := get_extended_material_designation_id_value(SELF);
  WHERE
    WR1:
END_ENTITY;

```

Attribute definitions:

id: the derived id attribute provided by an applied_identification_assignment.

5.2.3.11 extended_product_definition_shape

An extended_product_definition_shape is a product_definition_shape with a derived id attribute.

EXPRESS specification:

```

ENTITY extended_product_definition_shape
  SUBTYPE OF (product_definition_shape);
  DERIVED
    id: identifier := get_extended_product_definition_shape_id_value(SELF);
  WHERE
    WR1:
END_ENTITY;

```

Attribute definitions:

id: the derived id attribute provided by an applied_identification_assignment.

5.2.3.12 extended_product_related_product_category

An extended_product_related_product_category is a product_related_product_category with a derived id attribute.

EXPRESS specification:

```
ENTITY extended_product_related_product_category
  SUBTYPE OF (product_related_product_category);
  DERIVED
    id: identifier := get_extended_product_related_product_category_id_value(SELF);
  WHERE
    WR1:
END_ENTITY;
```

Attribute definitions:

id: the derived id attribute provided by an applied_identification_assignment.

5.2.3.13 extended_property_definition

An extended_property_definition is a property_definition with a derived id attribute.

EXPRESS specification:

```
ENTITY extended_property_definition
  SUBTYPE OF (property_definition);
  DERIVED
    id: identifier := get_extended_property_definition_id_value(SELF);
  WHERE
    WR1:
END_ENTITY;
```

Attribute definitions:

id: the derived id attribute provided by an applied_identification_assignment.

5.2.3.14 extended_shape_aspect

An `extended_shape_aspect` is a `shape_aspect` with a derived `id` attribute.

EXPRESS specification:

```
ENTITY extended_shape_aspect
  SUBTYPE OF (shape_aspect);
  DERIVED
    id: identifier := get_extended_shape_aspect_id_value(SELF);
  WHERE
    WR1:
END_ENTITY;
```

Attribute definitions:

id: the derived `id` attribute provided by an `applied_identification_assignment`.

5.2.3.15 extended_shape_aspect_relationship

An `extended_shape_aspect_relationship` is a `shape_aspect_relationship` with a derived `id` attribute.

EXPRESS specification:

```
ENTITY extended_shape_aspect_relationship
  SUBTYPE OF (shape_aspect_relationship);
  DERIVED
    id: identifier := get_extended_shape_aspect_relationship_id_value(SELF);
  WHERE
    WR1:
END_ENTITY;
```

Attribute definitions:

id: the derived `id` attribute provided by an `applied_identification_assignment`.

5.2.3.16 externally_defined_class

An `externally_defined_class` is a type of both `class` and `externally_defined_item`.

EXPRESS specification:

```
ENTITY externally_defined_class
  SUBTYPE OF (class, externally_defined_item);
END_ENTITY;
```

5.2.3.17 externally_defined_general_property

An externally_defined_general_property is a type of both general_property and externally_defined_item.

EXPRESS specification:

```
ENTITY externally_defined_general_property
  SUBTYPE OF (general_property, externally_defined_item);
END_ENTITY;
```

5.2.4 Ship structures imported entity modifications

5.2.4.1 compound_representation_item

The base definition of compound_representation_item is given in ISO 10303-43. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the compound_representation_item:

- comp_rep_item_w_name_eq_source (see 5.2.5.18)
- comp_rep_item_w_name_eq_source (see 5.2.5.22)
- comp_rep_item_w_name_eq_design_SWSF_values (see 5.2.5.35)
- comp_rep_item_w_name_eq_maximum_permissible_SWBM_values (see 5.2.5.36)
- comp_rep_item_w_name_eq_representations (see 5.2.5.69)
- comp_rep_item_w_name_eq_offset (see 5.2.5.144)
- comp_rep_item_w_name_eq_x_y_angle (see 5.2.5.145)

- comp_rep_item_w_name_eq_radius (see 5.2.5.147)
- comp_rep_item_w_name_eq_depth (see 5.2.5.149)
- comp_rep_item_w_name_eq_drain_hole_radius (see 5.2.5.150)
- comp_rep_item_w_name_eq_gap (see 5.2.5.151)
- comp_rep_item_w_name_eq_gap_radius (see 5.2.5.152)
- comp_rep_item_w_name_eq_width (see 5.2.5.153)
- comp_rep_item_w_name_eq_the_function (see 5.2.5.155)
- comp_rep_item_w_name_eq_half_axis_a (see 5.2.5.157)
- comp_rep_item_w_name_eq_half_axis_b (see 5.2.5.158)
- comp_rep_item_w_name_eq_distance (see 5.2.5.160)
- comp_rep_item_w_name_eq_r1 (see 5.2.5.161)
- comp_rep_item_w_name_eq_r2 (see 5.2.5.162)
- comp_rep_item_w_name_eq_bounding_curve (see 5.2.5.164)
- comp_rep_item_w_name_eq_the_function (see 5.2.5.166)
- comp_rep_item_w_name_eq_x_offset (see 5.2.5.168)
- comp_rep_item_w_name_eq_y_offset (see 5.2.5.169)
- comp_rep_item_w_name_eq_radius (see 5.2.5.170)
- comp_rep_item_w_name_eq_x_offset (see 5.2.5.172)
- comp_rep_item_w_name_eq_y_offset (see 5.2.5.173)
- comp_rep_item_w_name_eq_radius (see 5.2.5.174)
- comp_rep_item_w_name_eq_R_left (see 5.2.5.176)
- comp_rep_item_w_name_eq_R_right (see 5.2.5.177)
- comp_rep_item_w_name_eq_x_A (see 5.2.5.178)

- comp_rep_item_w_name_eq_x_B (see 5.2.5.179)
- comp_rep_item_w_name_eq_x_C (see 5.2.5.180)
- comp_rep_item_w_name_eq_x_D (see 5.2.5.181)
- comp_rep_item_w_name_eq_y_B (see 5.2.5.182)
- comp_rep_item_w_name_eq_y_C (see 5.2.5.183)
- comp_rep_item_w_name_eq_x_depth (see 5.2.5.185)
- comp_rep_item_w_name_eq_y_depth (see 5.2.5.186)
- comp_rep_item_w_name_eq_radius (see 5.2.5.187)
- comp_rep_item_w_name_eq_length_of (see 5.2.5.189)
- comp_rep_item_w_name_eq_width (see 5.2.5.190)
- comp_rep_item_w_name_eq_corner_radius (see 5.2.5.192)
- comp_rep_item_w_name_eq_length_of (see 5.2.5.193)
- comp_rep_item_w_name_eq_width (see 5.2.5.194)
- comp_rep_item_w_name_eq_edge_radius (see 5.2.5.196)
- comp_rep_item_w_name_eq_distance (see 5.2.5.197)
- comp_rep_item_w_name_eq_length_of (see 5.2.5.198)
- comp_rep_item_w_name_eq_width (see 5.2.5.199)
- comp_rep_item_w_name_eq_x_z_angle (see 5.2.5.201)
- comp_rep_item_w_name_eq_offset (see 5.2.5.202)
- comp_rep_item_w_name_eq_x_y_angle (see 5.2.5.203)
- comp_rep_item_w_name_eq_c2_x (see 5.2.5.205)
- comp_rep_item_w_name_eq_c3_x (see 5.2.5.206)
- comp_rep_item_w_name_eq_c3_y (see 5.2.5.207)

- comp_rep_item_w_name_eq_c1_radius (see 5.2.5.208)
- comp_rep_item_w_name_eq_c2_radius (see 5.2.5.209)
- comp_rep_item_w_name_eq_c3_radius (see 5.2.5.210)

5.2.4.2 product_definition

The base definition of product_definition is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_definition:

- PRODUCT_DEFINITION_w_class_id_design_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_still_water_bending_moment_parameters (see 5.2.5.29)
- PRODUCT_DEFINITION_w_class_id_design_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED. (see 5.2.5.30)
- PRODUCT_DEFINITION_w_class_id_design_vertical_wave_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_vertical_wave_bending_moment_parameters (see 5.2.5.31)
- PRODUCT_DEFINITION_w_class_id_design_vertical_wave_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_vertical_wave_shear_force_parameters (see 5.2.5.32)
- PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED. (see 5.2.5.33)
- PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED. (see 5.2.5.34)
- product_definition_w_class_id_global_axis_placement_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.47)
- product_definition_w_class_id_principal_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.73)

- product_definition_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.83)
- PRODUCT_DEFINITION_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_class_notation_via_DEFINITION (see 5.2.5.85)
- PRODUCT_DEFINITION_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_regulation_via_DEFINITION (see 5.2.5.86)
- product_definition_w_class_id_class_parameters_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.92)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_NAME_eq_IMO_number_or_pennant_hull_number (see 5.2.5.94)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_call_sign (see 5.2.5.95)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_flag_state (see 5.2.5.96)
- PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_port_of_registration (see 5.2.5.97)
- PRODUCT_DEFINITION_w_class_id_shipyard_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard (see 5.2.5.98)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company (see 5.2.5.99)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_ordering_company (see 5.2.5.100)
- PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_owning_company (see 5.2.5.101)
- product_definition_w_class_id_freeboard_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.102)

- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_DATE_AND_TIME_ASSIGNMENT_via_ITEMS_w_ROLE_eq_date_freeboard_assigned (see 5.2.5.103)
- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_assigned_by (see 5.2.5.104)
- PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_loadline_via_DEFINITION (see 5.2.5.106)
- product_definition_w_class_id_lightship_definition_referenced_by_exactly_1_property_definition_representation_via_property_definition (see 5.2.5.109)

5.2.4.3 product_definition_shape

The base definition of product_definition_shape is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_definition_shape:

- PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_manufacturing_definition_parameters (see 5.2.5.10)
- PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_definition_parameters (see 5.2.5.11)
- PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE (see 5.2.5.12)
- PRODUCT_DEFINITION_SHAPE_w_class_id_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters (see 5.2.5.16)
- PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE (see 5.2.5.17)
- PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE (see 5.2.5.19)

- PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_-
exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_refer-
enced_by_exactly_2_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.256)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_refer-
enced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_-
w_NAME_eq_corrugated_part_design_parameters (see 5.2.5.257)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_refer-
enced_by_exactly_1_SHAPE_ASPECT_w_class_id_corrugation_via_OF_SHAPE (see 5.2.5.258)
- PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_-
exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.267)
- PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_-
exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_-
eq_plate_design_parameters (see 5.2.5.268)
- PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_-
exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE (see 5.2.5.269)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_-
exactly_N_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.274)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_-
exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_-
eq_profile_design_parameters (see 5.2.5.275)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_-
exactly_1_SHAPE_ASPECT_w_class_id_profile_curve_trace_line_via_OF_SHAPE (see 5.2.5.276)
- PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_-
exactly_1_SHAPE_ASPECT_w_class_id_profile_cross_section_via_OF_SHAPE (see 5.2.5.277)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_-
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.292)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_-
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEF-
INITION_w_NAME_eq_corrugated_structure_design_parameters (see 5.2.5.293)
- PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_-
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE
(see 5.2.5.294)

- PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE (see 5.2.5.306)
- PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_design_parameters (see 5.2.5.307)
- PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE (see 5.2.5.308)
- PRODUCT_DEFINITION_SHAPE_w_class_id_structural_system_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_structural_system_design_parameters (see 5.2.5.323)
- PRODUCT_DEFINITION_SHAPE_w_class_id_beveled_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_beveled_groove_weld_design_parameters (see 5.2.5.325)
- PRODUCT_DEFINITION_SHAPE_w_class_id_butt_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_butt_groove_weld_design_parameters (see 5.2.5.328)
- PRODUCT_DEFINITION_SHAPE_w_class_id_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_fillet_weld_design_parameters (see 5.2.5.330)
- PRODUCT_DEFINITION_SHAPE_w_class_id_continuous_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_continuous_fillet_weld_design_parameters (see 5.2.5.331)
- PRODUCT_DEFINITION_SHAPE_w_class_id_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_groove_weld_design_parameters (see 5.2.5.334)
- PRODUCT_DEFINITION_SHAPE_w_class_id_intermittent_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_intermittent_fillet_weld_design_parameters (see 5.2.5.336)
- PRODUCT_DEFINITION_SHAPE_w_class_id_spot_seam_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_spot_seam_weld_design_parameters (see 5.2.5.338)
- PRODUCT_DEFINITION_SHAPE_w_class_id_weld_filler_material_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weld_filler_material_design_parameters (see 5.2.5.341)

- PRODUCT_DEFINITION_SHAPE_w_class_id_welded_joint_design_definition_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_welded_joint_design_parameters (see 5.2.5.343)
- PRODUCT_DEFINITION_SHAPE_w_class_id_structural_part_joint_design_definition_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_structural_part_joint_design_parameters (see 5.2.5.348)

5.2.4.4 property_definition

The base definition of property_definition is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the property_definition:

- PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system (see 5.2.5.51)
- PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_exactly_1_ITEM_DEFINED_TRANSFORMATION_via_TRANSFORM_ITEM_1_w_NAME_eq_transformation_to_parent (see 5.2.5.53)
- PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_with_position_reference_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w NAME_eq_local_co_ordinate_system_with_position_reference (see 5.2.5.56)
- PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w NAME_eq_class_notation (see 5.2.5.87)
- PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_class_society (see 5.2.5.88)
- PROPERTY_DEFINITION_w_class_id_loadline_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w NAME_eq_loadline (see 5.2.5.107)
- PROPERTY_DEFINITION_w_class_id_lightship_weight_item_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w NAME_eq_lightship_weight_item (see 5.2.5.111)
- PROPERTY_DEFINITION_w_class_id_weight_and_centre_of_gravity_referenced_by_ex-

actly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_-
eq_weight_and_centre_of_gravity (see 5.2.5.113)

- PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_-
eq_ship_material_property_parameters (see 5.2.5.118)
- PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE.NAME_-
eq_material_reference (see 5.2.5.120)
- PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_-
NAME_-eq_homogeneous_ship_material_property_parameters (see 5.2.5.121)
- PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_at_most_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE.NAME_-
eq_material_reference (see 5.2.5.123)
- PROPERTY_DEFINITION_w_class_id_plate_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_-
eq_plate_function_parameters (see 5.2.5.271)
- PROPERTY_DEFINITION_w_class_id_profile_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w NAME_-
eq_profile_function_parameters (see 5.2.5.279)
- PROPERTY_DEFINITION_w_class_id_corrugated_structure_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w NAME_-
eq_corrugated_structure_function_parameters (see 5.2.5.297)
- PROPERTY_DEFINITION_w_class_id_panel_system_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_-
NAME_-eq_panel_system_function_parameters (see 5.2.5.309)
- PROPERTY_DEFINITION_w_class_id_plate_stake_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_-
NAME_-eq_plate_stake_function_parameters (see 5.2.5.314)

5.2.4.5 property_definition_representation

The base definition of property_definition_representation is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the property_definition_representation:

- PROPERTY_DEFINITION_REPRESENTATION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_REPRESENTATION_via_USED_REPRESENTATION_w_NAME_eq_moment_3d (see 5.2.5.115)

5.2.4.6 representation

The base definition of representation is given in ISO 10303-43. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the representation:

- one_item_in_rep_if_used_in_pdr_w_name_eq_plate_renewal_definition_attributes (see 5.2.5.4)
- one_item_in_rep_if_used_in_pdr_w_name_eq_profile_renewal_definition_attributes (see 5.2.5.5)
- one_item_each_w_name_hull_cross_section_result_in_rep_of_class_actual_first_moment_horizontal (see 5.2.5.6)
- one_item_in_rep_if_used_in_pdr_w_name_eq_design_still_water_bending_moment_parameters (see 5.2.5.23)
- one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_moment_parameters (see 5.2.5.24)
- one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_parameters (see 5.2.5.25)
- one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_moment_parameters (see 5.2.5.26)
- one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_parameters (see 5.2.5.27)
- one_item_in_rep_if_used_in_pdr_w_name_eq_table of shear force value (see 5.2.5.28)
- one_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_definition_parameters (see 5.2.5.40)
- one_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters (see 5.2.5.43)

- one_item_in_rep_if_used_in_pdr_w_name_eq_global_axis_placement (see 5.2.5.48)
- one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system (see 5.2.5.52)
- one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system_with_position_ref-
erence (see 5.2.5.57)
- one_item_in_rep_if_used_in_pdr_w_name_eq_maximum_permissible_SWSF_values (see 5.2.5.70)
- one_item_in_rep_if_used_in_pdr_w_name_eq_principal_characteristics (see 5.2.5.74)
- one_item_in_rep_if_used_in_pdr_w_name_eq_class_and_statutory_designation (see 5.2.5.84)
- one_item_in_rep_if_used_in_pdr_w_name_eq_class_notation (see 5.2.5.91)
- one_item_in_rep_if_used_in_pdr_w_name_eq_class_parameters (see 5.2.5.93)
- one_item_in_rep_if_used_in_pdr_w_name_eq_freeboard_characteristics (see 5.2.5.105)
- one_item_in_rep_if_used_in_pdr_w_name_eq_loadline (see 5.2.5.108)
- one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_definition (see 5.2.5.110)
- one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_weight_item (see 5.2.5.112)
- one_item_in_rep_if_used_in_pdr_w_name_eq_weight_and_centre_of_gravity (see 5.2.5.114)
- one_item_in_rep_if_name_eq_moment_3d (see 5.2.5.116)
- one_item_in_rep_if_used_in_pdr_w_name_eq_ship_material_property_parameters (see 5.2.5.119)
- one_item_in_rep_if_used_in_pdr_w_name_eq_homogeneous_ship_material_property_para-
meters (see 5.2.5.122)
- one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_definition_parame-
ters (see 5.2.5.214)
- one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_position_parame-
ters (see 5.2.5.216)
- one_item_in_rep_if_used_in_pdr_w_name_eq_structural_added_material_feature_design-
definition_parameter_representation (see 5.2.5.220)
- one_item_in_rep_if_used_in_pdr_w_name_eq_weld_manufacturing_definition_parameters
(see 5.2.5.223)

- one_item_in_rep_if_used_in_pdr_w_name_eq_welding_sequence_representation (see 5.2.5.225)
- one_item_in_rep_if_used_in_pdr_w_name_eq_angle_bar_cross_section_design_parameters (see 5.2.5.231)
- one_item_in_rep_if_used_in_pdr_w_name_eq_bulb_flat_cross_section_design_parameters (see 5.2.5.232)
- one_item_in_rep_if_used_in_pdr_w_name_eq_channel_cross_section_design_parameters (see 5.2.5.233)
- one_item_in_rep_if_used_in_pdr_w_name_eq_circular_hollow_profile_cross_section_design_parameters (see 5.2.5.234)
- one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_part_design_parameters (see 5.2.5.235)
- one_item_in_rep_if_used_in_pdr_w_name_eq_corrugation_design_parameters (see 5.2.5.237)
- one_item_in_rep_if_used_in_pdr_w_name_eq_explicit_profile_cross_section_shape (see 5.2.5.239)
- one_item_in_rep_if_used_in_pdr_w_name_eq_flanged_plate_cross_section_design_parameters (see 5.2.5.240)
- one_item_in_rep_if_used_in_pdr_w_name_eq_flat_bar_cross_section_design_parameters (see 5.2.5.241)
- one_item_in_rep_if_used_in_pdr_w_name_eq_plate_design_parameters (see 5.2.5.242)
- one_item_in_rep_if_used_in_pdr_w_name_eq_profile_curve_trace_line_design_parameters (see 5.2.5.245)
- one_item_in_rep_if_used_in_pdr_w_name_eq_profile_design_parameters (see 5.2.5.247)
- one_item_in_rep_if_used_in_pdr_w_name_eq_round_bar_cross_section_design_parameters (see 5.2.5.250)
- one_item_in_rep_if_used_in_pdr_w_name_eq_section_properties (see 5.2.5.251)
- one_item_in_rep_if_used_in_pdr_w_name_eq_square_tube_cross_section_design_parameters (see 5.2.5.252)
- one_item_in_rep_if_used_in_pdr_w_name_eq_twist_location_design_parameters (see 5.2.5.253)
- one_item_in_rep_if_used_in_pdr_w_name_eq_t_bar_cross_section_design_parameters (see 5.2.5.254)

- one_item_in_rep_if_used_in_pdr_w_name_eq_w_shape_cross_section (see 5.2.5.255)
- one_item_in_rep_if_used_in_pdr_w_name_eq_plate_function_parameters (see 5.2.5.270)
- one_item_in_rep_if_used_in_pdr_w_name_eq_profile_function_parameters (see 5.2.5.278)
- one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_symmetry_relationship_design_parameters (see 5.2.5.283)
- one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_design_parameters (see 5.2.5.295)
- one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_function_parameters (see 5.2.5.299)
- one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_curve_boundary_design_parameters (see 5.2.5.303)
- one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_design_parameters (see 5.2.5.304)
- one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_function_parameters (see 5.2.5.310)
- one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_plane_boundary_design_parameters (see 5.2.5.313)
- one_item_in_rep_if_used_in_pdr_w_name_eq_plate_strake_function_parameters (see 5.2.5.315)
- one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_design_parameters (see 5.2.5.319)
- one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_symmetry_relationship_design_parameters (see 5.2.5.321)
- one_item_in_rep_if_used_in_pdr_w_name_eq_beveled_groove_weld_design_parameters (see 5.2.5.326)
- one_item_in_rep_if_used_in_pdr_w_name_eq_butt_groove_weld_design_parameters (see 5.2.5.329)
- one_item_in_rep_if_used_in_pdr_w_name_eq_electrode_chemical_composition_design_parameters (see 5.2.5.332)
- one_item_in_rep_if_used_in_pdr_w_name_eq_fillet_weld_design_parameters (see 5.2.5.333)
- one_item_in_rep_if_used_in_pdr_w_name_eq_groove_weld_design_parameters (see 5.2.5.335)
- one_item_in_rep_if_used_in_pdr_w_name_eq_intermittent_fillet_weld_design_parameters (see 5.2.5.337)

- `one_item_in_rep_if_used_in_pdr_w_name_eq_spot_seam_weld_design_parameters` (see 5.2.5.339)
- `one_item_in_rep_if_used_in_pdr_w_name_eq_weld_design_parameters` (see 5.2.5.340)
- `one_item_in_rep_if_used_in_pdr_w_name_eq_welded_joint_design_parameters` (see 5.2.5.344)
- `one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_joint_design_parameters` (see 5.2.5.350)
- `one_item_in_rep_if_used_in_pdr_w_name_eq_continuous_fillet_weld_design_parameters` (see 5.2.5.351)
- `one_or_more_items_in_rep_if_used_in_pdr_w_name_eq_class_notation` (see 5.2.5.89)
- `rep_has_guac` (see 5.2.5.3)

5.2.4.7 representation_item

The base definition of `representation_item` is given in ISO 10303-43. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the `representation_item`:

- `dsc_rep_item_w_n_user_def_function_if_dsc_rep_item_w_n_function_and_d_user_defined`
- `dsc_rep_item_w_n_user_defined_tightness_if_dsc_rep_item_w_n_tightness_and_d_user_defined`

5.2.5 Ship structures rule definitions

5.2.5.1 `mandatory_approvals_w_role_approval_history_for_attr_approval_for_each_applied_approval_assignment_w_role_subject_for_attr_assigned_approval`

The `mandatory_approvals_w_role_approval_history_for_attr_approval_for_each_applied_approval_assignment_w_role_subject_for_attr_assigned_approval` specifies that for each instance of T1 (...assignment entity) with role R1 that references through attribute A1 an instance of type T3 there shall be another ...assignment instance of type T2 and with role R2 that references T3 via attribute A2. `applied_approval_assignment`: T1 - entity type of the referencing instance; 'subject': R1 - role of the referencing instance; `assigned_approval`: A1 - name of the attribute that T1 references T3 with; 'approvals': T2 - entity type of the referencing instance that shall exist if there is a T1 for T3; 'approval_history': R2 -

role of the mandatory referencing instance; approval: A2 - name of the attribute that T2 references T3 with; role: T3 - entity type of the referenced instance. *)

EXPRESS specification:

```

RULE mandatory_ 'approvals' _w_role_approval_history_for_attr_approval_for_
each_applied_approval_assignment_w_role_subject_for_attr_assigned_approval
FOR (applied_approval_assignment);

LOCAL
    t1_set: SET OF applied_approval_assignment := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

t1_set := QUERY(all_t1_inst <* applied_approval_assignment |
                all_t1_inst.role.name = 'subject');

REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(QUERY(t2_inst <*
        USEDIN(t1_set[i].assigned_approval, 'SHIP_STRUCTURES.' +
        'approvals' + '.' + 'approval') |
        t2_inst.role.name = 'approval_history')) = 0;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.2 applied_group_assignment_with_role_approvals_has_at_least_1_and_-only_approval_in_items

The `applied_group_assignment_with_role_approvals_has_at_least_1_and_only_approval_in_items` forces the aggregate type attribute A of "assignment" entity type T1 with the object_role.name = R1 to have at least N elements all of which shall be of type T2. `applied_group_assignment`: T1 - type having attribute A; `items`: A1 - aggregate valued attribute of T1; 1: N - number of elements that A shall exactly have for all instances of T1; `'approvals'`: R1 - string containing the name of the role that T1 is playing; `approval`: T2 - mandatory entity type in T1.A1.

EXPRESS specification:

```

RULE applied_group_assignment_with_role_approvals_has_at_least_1_and_only_
approval_in_items FOR (applied_group_assignment);

WHERE

```



```

WR1: SIZEOF(QUERY(ass_inst <* applied_group_assignment |
    NOT((ass_inst.role.name = 'approvals')
        AND
            (SIZEOF(ass_inst.items) >= 1)
            AND
            (SIZEOF(QUERY(item <* ass_inst.items |
                NOT('approval' IN TYPEOF(item)))) = 0)
        )
    )) = 0;
END_RULE;

```

5.2.5.3 rep_has_guac

The rep_has_guac ensures that a representation has a global_unit_assigned_context if it has representation_items of type value_representation_item and/or geometric_representation_item.

EXPRESS specification:

```

RULE rep_has_guac FOR (REPRESENTATION);

LOCAL
    v_r_i:    STRING := 'SHIP_STRUCTURES.VALUE_REPRESENTATION_ITEM';
    g_r_i:    STRING := 'SHIP_STRUCTURES.GEOMETRIC_REPRESENTATION_ITEM';
    g_u_a_c:  STRING := 'SHIP_STRUCTURES.GLOBAL_UNIT_ASSIGNED_CONTEXT';
    has_guac: BOOLEAN := TRUE;
END_LOCAL;

REPEAT i := 1 TO HIINDEX(REPRESENTATION) WHILE has_guac;
    REPEAT j := 1 TO SIZEOF(REPRESENTATION[i].ITEMS) WHILE has_guac;
        IF ((v_r_i IN TYPEOF(REPRESENTATION[i].ITEMS[j])) OR
            (g_r_i IN TYPEOF(REPRESENTATION[i].ITEMS[j]))) THEN
            has_guac := (g_u_a_c IN TYPEOF(REPRESENTATION[i].CONTEXT_OF_ITEMS));
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: has_guac;
END_RULE;

```

5.2.5.4 one_item_in_rep_if_used_in_pdr_w_name_eq_plate_renewal_definition_attributes

The one_item_in_rep_if_used_in_pdr_w_name_eq_plate_renewal_definition_attributes forces the 'items' attribute of a representation to have for each entry in the list ('...') ex-

actly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'plate renewal definition attributes'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_plate_renewal_definition_
  attributes FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := 'rule thickness';
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'plate renewal definition
        attributes')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.5 one_item_in_rep_if_used_in_pdr_w_name_eq_profile_renewal_ definition_attributes

The one_item_in_rep_if_used_in_pdr_w_name_eq_profile_renewal_definition_attributes forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'profile renewal definition attributes'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_profile_renewal_definition_
  attributes FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := 'renewal modulus';
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'profile renewal definition
        attributes')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.6 one_item_each_w_name_hull_cross_section_result_in_rep_of_class- actual_first_moment_horizontal

The one_item_each_w_name_hull_cross_section_result_in_rep_of_class_actual_first_moment_horizontal Forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list, if the representation is of class 'hull cross section result'.

EXPRESS specification:

```

RULE one_item_each_w_name_hull_cross_section_result_in_rep_of_class_actual_
  first_moment_horizontal FOR (representation);

```

```
...
END_RULE;
```

5.2.5.7 id_is_globally_unique

The `id_is_globally_unique` forces the globally uniqueness of the attribute 'id' of all definable_objects (e.g. those being classified to be a 'definable object').

EXPRESS specification:

```
RULE id_is_globally_unique FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
    duplicate:      BOOLEAN := FALSE;
    def_obj_cl_ass: SET OF CLASSIFICATION_ASSIGNMENT := [];
    clfied_inst:    SET OF GENERIC := [];
END_LOCAL;

def_obj_cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
                        i.assigned_classification.name = 'definable object');

REPEAT i := 1 TO HIINDEX(def_obj_cl_ass);
    clfied_inst := clfied_inst + def_obj_cl_ass[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(clfied_inst) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(clfied_inst) WHILE NOT duplicate;
        IF (clfied_inst[i].id = clfied_inst[j].id) THEN
            duplicate := TRUE;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: NOT(duplicate);
END_RULE;
```

5.2.5.8 ID_of_extended_property_definition_with_class_versionable_object_matches_pattern

The `ID_of_extended_property_definition_with_class_versionable_object_matches_pattern` forces string type attribute A of type T to match the pattern P if the instance of T is of class C.
extended_property_definition: T - type having attribute A; 'versionable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_extended_property_definition_with_class_versionable_object_
  pattern FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
    'versionable object') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
END_RULE;

```

5.2.5.9 ID_of_representation_with_class_versionable_object_matches_pattern

The ID_of_representation_with_class_versionable_object_matches_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. representation: T - type having attribute A; 'versionable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_representation_with_class_versionable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
    'versionable object') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
END_RULE;

```

5.2.5.10 PRODUCT_DEFINITION_SHAPE_w_class_id_library_- manufacturing_definition_referenced_by_exactly_1_SHAPE_- ASPECT_via_OF_SHAPE_w_NAME_eq_library_manufacturing_- definition_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_manufacturing_definition_parameters gets all classification_assignment instances with id 'library manufacturing definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose attribute 'library manufacturing definition parameters' has the value " check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_
referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_
manufacturing_definition_parameters
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'library manufacturing
                definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
                  'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.'library manufacturing
                  definition parameters' = )) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.11 PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_definition_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_definition_parameters gets all classification_assignment instances with id 'library definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose attribute 'library definition parameters' has the value library_definition_parameters check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_
  exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_definition_parameters
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'library definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
    'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME =
    'library definition parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.12 PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE gets all classification_assignment instances with id 'library manufacturing definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'source' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set:  SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'library manufacturing
                definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
                    'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set | 'source' IN WHICH_CLASS(i)))
                    < 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```


5.2.5.13 ID_of_extended_product_definition_shape_with_class_versionable-object_matches_pattern

The ID_of_extended_product_definition_shape_with_class_versionable-object_matches_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C; extended_product_definition_shape: T - type having attribute A; 'versionable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match

EXPRESS specification:

```

RULE ID_of_extended_product_definition_shape_with_class_versionable_object_
  pattern FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
    'versionable object') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
END_RULE;

```

5.2.5.14 library_definition_has_max_items_elements_in_1

The library_definition_has_max_items_elements_in_1 forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE library_definition_has_max_items_elements_in_1 FOR (representation);

LOCAL
  var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
  temp1 <* representation |
    SIZEOF (
      QUERY(

```

```

        temp2 <* USEDIN(temp1,
'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
| ((temp2\classification_assignment.role.name =
    'class membership') AND
    (temp_appl_class\classification_assignment.
      assigned_classification\group.name = 'library definition'))
    )
    ) > 0
);
WHERE
  wr1: SIZEOF(QUERY(temp3 <* var_representation |
    SIZEOF(temp3.1) > items)) = 0;
END_RULE;

```

5.2.5.15 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
  FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

  LOCAL
    version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects: BAG OF GENERIC := [];
    duplicate:        BOOLEAN := FALSE;
  END_LOCAL;

  version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
    i.ROLE.NAME = 'version identifier');

  REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
  END_REPEAT;

  REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
      duplicate := versionable_objects[i] :=: versionable_objects[j];
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT duplicate;
END_RULE;

```

5.2.5.16 PRODUCT_DEFINITION_SHAPE_w_class_id_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters gets all classification_assignment instances with id 'manufacturing definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose attribute NAME has the value "manufacturing definition parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_manufacturing_definition_referenced_
by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);
  LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME =
      'manufacturing definition');

  REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
      t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
  END_REPEAT;

  REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
      'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
      i.NAME = 'manufacturing definition parameters')) = 1);
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.17 PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE gets all classification_assignment instances with id 'library definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'source' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_
  exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'library definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
    'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | 'source' IN WHICH_CLASS(i)))
    < 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.18 comp_rep_item_w_name_eq_source

The comp_rep_item_w_name_eq_source forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'source': VAL - value of the inherited name attribute of the compound_representation_item; 'library definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```
RULE comp_rep_item_w_name_eq_source FOR (compound_representation_item);
```

```

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'library definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
               temp_comp_rep_item.name = 'library definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
      violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;
```

5.2.5.19 PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_design_parameters gets the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose attribute NAME has the value 'library design parameters' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_
  by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'library design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
    'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'library design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.20 library_design_definition_has_max_items_elements_in_1

The library_design_definition_has_max_items_elements_in_1 forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE library_design_definition_has_max_items_elements_in_1 FOR (representation);

LOCAL
  var_representation: BAG OF representation := [];
END_LOCAL;

```

```

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
            | ((temp2\classification_assignment.role.name =
                'class membership') AND
                (temp_appl_class\classification_assignment.assigned_
                classification\group.name = 'library design definition')))) > 0
        );
WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.1) > items)) = 0;
END_RULE;

```

5.2.5.21 PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE gets all classification_assignment instances with id 'library design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'source' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_
by_exactly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'library design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;

```

```

END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
    'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | 'source' IN WHICH_CLASS(i)))
    < 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.22 comp_rep_item_w_name_eq_source

The `comp_rep_item_w_name_eq_source` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'source': VAL - value of the inherited name attribute of the `compound_representation_item`. 'library design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_source FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'library design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name = 'library design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
      rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```


5.2.5.23 one_item_in_rep_if_used_in_pdr_w_name_eq_design_still_water_bending_moment_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_design_still_water_bending_moment_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL 'design still water bending moment parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_design_still_water_bending_moment_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['hogging amidship', 'sagging amidship'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'design still water bending
                    moment parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.24 one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_moment_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_moment_pa-

rameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'design vertical wave bending moment parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_
moment_parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['hogging', 'sagging'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'design vertical wave bending
        moment parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.25 one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_- shear_force_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'design vertical wave shear force parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['negative value', 'positive value'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'design vertical wave shear
        force parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.26 one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave- bending_moment_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_moment_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'design vertical wave bending moment parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_
moment_parameters FOR (representation);

```

```

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := 'hogging', 'sagging';
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'design vertical wave bending
        moment parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.27 one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'design vertical wave shear force parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_shear_force_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := 'negative value', 'positive value';
  violation: BOOLEAN := FALSE;

```

```

END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name = 'design vertical wave shear
              force parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                   rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.28 one_item_in_rep_if_used_in_pdr_w_name_eq_table of shear force value

The one_item_in_rep_if_used_in_pdr_w_name_eq_table of shear force value forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'table of shear force value'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_table of shear force value
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := 'design SWSF values';
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (

```

```

        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name = 'table of shear force value')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.29 **PRODUCT_DEFINITION_w_class_id_design_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_still_water_bending_moment_parameters**

The **PRODUCT_DEFINITION_w_class_id_design_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_still_water_bending_moment_parameters** gets all classification assignment instances with id 'design still water bending moment' for all instances of **PRODUCT_DEFINITION** in **t1_set**: get the **PROPERTY_DEFINITION_REPRESENTATION** instances that are referencing a **PRODUCT_DEFINITION** instance via **DEFINITION**, filter out those **PROPERTY_DEFINITION_REPRESENTATION** instances whose attribute **NAME** has the value "design still water bending moment parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_design_still_water_bending_moment_referenced_
by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
design_still_water_bending_moment_parameters
FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION := [];
    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;

```

```

END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'design still water bending
                 moment');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'design still
  water bending moment parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.30 **PRODUCT_DEFINITION_w_class_id_design_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_.UNUSED.**

The **PRODUCT_DEFINITION_w_class_id_design_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_.UNUSED.** gets all classification_assignment instances with id 'design still water shear force' for all instances of **PRODUCT_DEFINITION** in t1_set: get the **PROPERTY_DEFINITION_REPRESENTATION** instances that are referencing a **PRODUCT_DEFINITION** instance via **DEFINITION**, filter out those **PROPERTY_DEFINITION_REPRESENTATION** instances whose attribute **NAME** has the value ".UNUSED." check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_design_still_water_shear_force_referenced_
by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
.UNUSED.
FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];

```

```

    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'design still water shear
                force');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
    REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = '.UNUSED.')) = 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.31 **PRODUCT_DEFINITION_w_class_id_design_vertical_wave_- bending_moment_referenced_by_exactly_1_PROPERTY_- DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_- eq_design_vertical_wave_bending_moment_parameters**

The PRODUCT_DEFINITION_w_class_id_design_vertical_wave_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_vertical_wave_bending_moment_parameters gets all classification_assignment instances with id 'design vertical wave bending moment' for all instances of PRODUCT_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "design vertical wave bending moment parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_design_vertical_wave_bending_moment_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_
NAME_eq_design_vertical_wave_bending_moment_parameters
FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

```



```

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION := [];
  t2_set:  SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'design vertical wave
                 bending moment');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'design vertical wave
  bending moment parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.32 **PRODUCT_DEFINITION_w_class_id_design_vertical_wave_shear_- force_referenced_by_exactly_1_PROPERTY_DEFINITION_- REPRESENTATION_via_DEFINITION_w_NAME_eq_design_- vertical_wave_shear_force_parameters**

The PRODUCT_DEFINITION_w_class_id_design_vertical_wave_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_design_vertical_wave_shear_force_parameters gets all classification_assignment instances with id 'design vertical wave shear force' for all instances of PRODUCT_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "design vertical wave shear force parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_design_vertical_wave_shear_force_referenced_
  by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  design_vertical_wave_shear_force_parameters

```

```

FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'design vertical wave
                 shear force');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'design vertical
  wave shear force parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.33 PRODUCT_DEFINITION_w_class_id_maximum_permmissible_still_ water_bending_moment_referenced_by_exactly_1_PROPERTY_ DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_ eq_.UNUSED.

The PRODUCT_DEFINITION_w_class_id_maximum_permmissible_still_water_bending_moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_.UNUSED. gets all classification_assignment instances with id 'maximum permissible still water bending moment' for all instances of PRODUCT_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value ".UNUSED." check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_bending_
moment_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_
DEFINITION_w_NAME_eq_.UNUSED.
FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'maximum permissible
  still water bending moment');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = '.UNUSED.')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.34 PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_ water_shear_force_referenced_by_exactly_1_PROPERTY_ DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_ eq_

The PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_shear_force_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_ gets all classification_assignment instances with id 'maximum permissi-
ble still water shear force' for all instances of PRODUCT_DEFINITION in t1_set: get
the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a
PRODUCT_DEFINITION instance via DEFINITION, filter out those PROPERTY_DE-
FINITION_REPRESENTATION instances whose attribute NAME has the value ".UN-
USED." check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_maximum_permmissible_still_water_shear_force_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_
NAME_eq_.UNUSED.
FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'maximum permmissible still water shear force');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = '.UNUSED.')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.35 comp_rep_item_w_name_eq_design_SWSF_values

The comp_rep_item_w_name_eq_design_SWSF_values forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. design_SWSF_values: VAL - value of the inherited name attribute of the compound_representation_item. 'negative value', 'positive value', 'position': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_design_SWSF_values

```

```

FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'negative value', 'positive value', 'position';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'design SWSF values');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.36 comp_rep_item_w_name_eq_maximum_permissible_SWBM_values

The comp_rep_item_w_name_eq_maximum_permissible_SWBM_values forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. maximum_permissible_SWBM_values: VAL - value of the inherited name attribute of the compound_representation_item. 'hogging', 'sagging', 'position': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_maximum_permissible_SWBM_values
FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'hogging', 'sagging', 'position';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'hogging', 'sagging',

```

```

        'position');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
            rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.37 COMPONENTS_RELATED_TO_ADDRESS

The COMPONENTS_RELATED_TO_ADDRESS Ensure, that each element in the parameter list is used in the description attribute of an external_source which shall be related to another external_source of class 'address'.

EXPRESS specification:

```

RULE COMPONENTS_RELATED_TO_ADDRESS FOR (EXTERNAL_SOURCE);

LOCAL
    parm_list: LIST OF STRING := ['internal location', 'street number',
        'street', 'postal box', 'town', 'region', 'postal code', 'country',
        'facsimile number', 'telephone number', 'electronic mail address',
        'telex number'];
    addr_list: SET OF EXTERNAL_SOURCE;
    comp_list: SET OF EXTERNAL_SOURCE;
    addr_relationships: SET OF EXTERNAL_SOURCE_RELATIONSHIP := [];
    comp_relationships: SET OF EXTERNAL_SOURCE_RELATIONSHIP := [];
    found: BOOLEAN := FALSE;
END_LOCAL;

addr_list = QUERY(temp <* EXTERNAL_SOURCE
    | 'ADDRESS' IN WHICH_CLASS(temp));

REPEAT i:=1 TO HIINDEX(addr_list);
    addr_relationships := addr_relationships + USEDIN(addr_list[i],
        'SHIP_STRUCTURES.EXTERNAL_SOURCE_RELATIONSHIP.RELATED_SOURCE');
END_REPEAT;

REPEAT i:=1 TO HIINDEX(parm_list) WHILE (NOT found);
    comp_list := QUERY(temp <* EXTERNAL_SOURCE
        | temp.description = parm_list[i] );

```

```

END_REPEAT;

search whether there is an external_source_relationship that relates an external_sources as image of a

REPEAT j:=1 TO HIINDEX(comp_list) WHILE (NOT found);
    comp_relationships := USEDIN(comp_list[j],
        'SHIP_STRUCTURES.EXTERNAL_SOURCE_RELATIONSHIP.RELATING_SOURCE');
    found := SIZEOF(addr_relationships * comp_relationships) = 0 ;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.38 mandatory_product_definition_structural_system_in_product_definition_hull_cross_section

The mandatory_product_definition_structural_system_in_product_definition_hull_cross_section A product_definition of class C_PD1 that also is a group (group_assignment.role.name = 'equivalence') (typically an item_structure in the ARM) shall - if not empty - via the group reference at least one product_definition of class C_PD2. 'hull cross section': C_PD1 - the class of a product_definition that is linked to a group; 'structural system': C_PD2 - the class of another product_definition at least one of which shall be referenced by 'hull cross section'.

EXPRESS specification:

```

RULE mandatory_product_definition_structural_system_in_product_definition_hull_
    cross_section FOR (CLASSIFICATION_ASSIGNMENT, GROUP_ASSIGNMENT);

LOCAL
    cl_ass:    SET OF CLASSIFICATION_ASSIGNMENT := [];
    pd:        SET OF PRODUCT_DEFINITION := [];
    gr_ass:    SET OF GROUP_ASSIGNMENT := [];
    groups:    SET OF GROUP := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
    (i.role.name = 'class membership') AND
    (i.assigned_classification.name = 'hull cross section'));

REPEAT i := 1 TO HIINDEX(cl_ass);
    pd := pd + i.items;
END_REPEAT;

get the groups associated with these product_definitions

```

```

gr_ass := QUERY(i <* GROUP_ASSIGNMENT |
                i.role.name = 'equivalence');

REPEAT i := 1 TO HIINDEX(gr_ass);
  REPEAT j := 1 TO HIINDEX(gr_ass[i].items); -- should always be just one
    IF (gr_ass[i].items[j] IN pd) THEN
      groups := groups + gr_ass.assigned_group;
    END_IF;
  END_REPEAT;
END_REPEAT;

gr_ass := QUERY(i <* GROUP_ASSIGNMENT |
                (SIZEOF(i.items) <> 0) AND
                (i.role.name = 'item structure') AND
                (i.assigned_group IN groups));

REPEAT i := 1 TO HIINDEX(gr_ass) WHILE NOT violation;
  violation := SIZEOF(QUERY(inst <* gr_ass[i].items |
                           'structural system' IN WHICH_CLASS(inst))) = 0;
END_REPEAT;

WHERE
  WR1: NOT violation;
END_RULE;

```

5.2.5.39 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
  FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

  LOCAL
    version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects: BAG OF GENERIC := [];
    duplicate:        BOOLEAN := FALSE;
  END_LOCAL;

  version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
                      i.ROLE.NAME = 'version identifier');

  REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
  END_REPEAT;

```



```

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
  REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    duplicate := versionable_objects[i] :=: versionable_objects[j];
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT duplicate;
END_RULE;

```

5.2.5.40 one_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_- design_definition_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_definition_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'hull cross section design definition parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_
  definition_parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['mirrored symmetry'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'hull cross section design
        definition parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;

```

```

END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.41 1_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_definition_parameters

The 1_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_definition_parameters forces the 'items' attribute of a representation that is the 'used_representation' in a property_definition_representation whose 'name' attribute has the value VAL to have exactly N representation_items. 'hull cross section design definition parameters': VAL - value of the .name attribute of a property_definition_representation; 1: N - maximum number of representation_items collected by the current representation.

EXPRESS specification:

```

RULE 1_item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_
    definition_parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'hull cross section design
                    definition parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    violation := SIZEOF(reps[i].items) > 1;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.42 shape_aspect_w_class_id_flare_area_referenced_by_exactly_2- property_definition_representation_via_definition_w_name_eq_flare- area_parameters

The shape_aspect_w_class_id_flare_area_referenced_by_exactly_2_property_definition_representation_via_definition_w_name_eq_flare_area_parameters gets all classification_assignment instances with id 'flare area' for all instances of shape_aspect in t1_set: get the property_definition_representation instances that are referencing a shape_aspect instance via definition, filter out those property_definition_representation instances whose attribute name has the value "flare area parameters" check if their number equals 2.

EXPRESS specification:

```

RULE shape_aspect_w_class_id_flare_area_referenced_by_exactly_2_property_
  definition_representation_via_definition_w_name_eq_flare_area_parameters
  FOR(shape_aspect, property_definition_representation, APPLIED_CLASSIFICATION_
  ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF shape_aspect := [];
  t2_set: SET OF property_definition_representation := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'flare area');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'property_definition_representation' + '.' + 'definition');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.name =
    'flare area parameters')) = 2);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.43 one_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL 'flare area parameters': VAL - value of the .name attribute of the property_definition_representation; 'area', 'height': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['area', 'height'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'flare area parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.44 2_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters

The `2_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters` forces the 'items' attribute of a representation that is the 'used_representation' in a property_definition_rep-

representation whose 'name' attribute has the value VAL to have exactly N representation_items. 'flare area parameters': VAL - value of the .name attribute of a property_definition_representation; 2: N - maximum number of representation_items collected by the current representation.

EXPRESS specification:

```

RULE 2_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'flare area parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    violation := SIZEOF(reps[i].items) > 2;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.45 ID_of_PRODUCT_with_class_definable_object_matches_pattern

The ID_of_PRODUCT_with_class_definable_object_matches_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. PRODUCT: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; *.*: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_PRODUCT_with_class_definable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

```

```

LOCAL
  clified_inst : SET OF GENERIC := [];
END_LOCAL;

  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
    'definable object') THEN
    clified_inst := clified_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clified_inst | NOT(i.ID LIKE *))) = 0;
END_RULE;

```

5.2.5.46 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object FOR(APPLIED_IDENTIFICATION_
  ASSIGNMENT);

LOCAL
  version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  versionable_objects: BAG OF GENERIC := [];
  duplicate:        BOOLEAN := FALSE;
END_LOCAL;

version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
  i.ROLE.NAME = 'version identifier');

REPEAT i := 1 TO HIINDEX(version_ids);
  versionable_objects := versionable_objects + version_ids[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
  REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    duplicate := versionable_objects[i] :=: versionable_objects[j];
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT duplicate;
END_RULE;

```

5.2.5.47 product_definition_w_class_id_global_axis_placement_referenced_by_exactly_1_property_definition_representation_via_property_definition

The product_definition_w_class_id_global_axis_placement_referenced_by_exactly_1_property_definition_representation_via_property_definition forces that a product_definition with a class id C is referenced by one property_definition_representation with the name C via a property_definition. 'global axis placement': C - class.name and property_definition_representation.name

EXPRESS specification:

```
RULE product_definition_w_class_id_global_axis_placement_referenced_by_exactly_
  1_property_definition_representation_via_property_definition
  FOR (product_definition, property_definition, property_definition_
  representation, applied_classification_assignment);
```

```
LOCAL
```

```
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF product_definition := [];
  t2_set:  SET OF property_definition_representation := [];
  t3_set:  SET OF property_definition := [];
  t4_set:  SET OF product_definition := [];
```

```
  violation: BOOLEAN := FALSE;
```

```
END_LOCAL;
```

```
c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'global axis placement');
```

```
REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;
```

```
t2_set:= QUERY(i <* PROPERTY_DEFINITION_REPRESENTATION |
  i.NAME = 'global axis placement');
```

```
REPEAT i := 1 TO HIINDEX(t2_set);
  t3_set := t3_set + t2_set[i].definition;
END_REPEAT;
```

```
REPEAT i := 1 TO HIINDEX(t3_set);
  t4_set := t4_set + t3_set[i].definition;
END_REPEAT;
```

```

violation := t1_set <> t4_set;

WHERE
    wr1: NOT violation;

END_RULE;

```

5.2.5.48 one_item_in_rep_if_used_in_pdr_w_name_eq_global_axis_placement

The `one_item_in_rep_if_used_in_pdr_w_name_eq_global_axis_placement` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'global axis placement': VAL - value of the .name attribute of the property_definition_representation; 'global axes and origin', 'after perpendicular offset', 'orientation': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_global_axis_placement
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['global axes and origin',
                                'after perpendicular offset', 'orientation'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'global axis placement')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;

```



```

END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.49 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
    FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

    LOCAL
        version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
        versionable_objects: BAG OF GENERIC := [];
        duplicate:        BOOLEAN := FALSE;
    END_LOCAL;

    version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
        i.ROLE.NAME = 'version identifier');

    REPEAT i := 1 TO HIINDEX(version_ids);
        versionable_objects := versionable_objects + version_ids[i].items;
    END_REPEAT;

    REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
        REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
            duplicate := versionable_objects[i] :=: versionable_objects[j];
        END_REPEAT;
    END_REPEAT;

    WHERE
        wr1: NOT duplicate;
END_RULE;

```

5.2.5.50 id_of_product_definition_with_class_global_axis_placement_matches_pattern

The id_of_product_definition_with_class_global_axis_placement_matches_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. product_definition: T - type having attribute A; 'global axis placement': C - class of T; id: A

- attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_definition_with_class_global_axis_placement_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'global axis placement') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:))) = 0;
END_RULE;

```

5.2.5.51 PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system

The PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system specifies all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "local co ordinate system" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_
  co_ordinate_system
  FOR (PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

  LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];

```

```

    t1_set: SET OF PROPERTY_DEFINITION := [];
    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'local co ordinate system');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                   i.NAME = 'local co ordinate system')) = 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.52 one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system

The `one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'local co ordinate system': VAL - value of the .name attribute of the property_definition_representation; 'local axes and origin': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['local axes and origin'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

```

```

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name = 'local co ordinate system')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.53 PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_-referenced_by_exactly_1_ITEM_DEFINED_TRANSFORMATION_via_TRANSFORM_ITEM_1_w_NAME_eq_transformation_to_parent

The PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_exactly_1_ITEM_DEFINED_TRANSFORMATION_via_TRANSFORM_ITEM_1_w_NAME_eq_transformation_to_parent specifies all instances of PROPERTY_DEFINITION in t1_set: get the ITEM_DEFINED_TRANSFORMATION instances that are referencing a PROPERTY_DEFINITION instance via TRANSFORM_ITEM_1, filter out those ITEM_DEFINED_TRANSFORMATION instances whose attribute NAME has the value "transformation to parent" check if their number equals 1

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_referenced_by_
    exactly_1_ITEM_DEFINED_TRANSFORMATION_via_TRANSFORM_ITEM_1_w_NAME_eq_
    transformation_to_parent
    FOR(PROPERTY_DEFINITION, ITEM_DEFINED_TRANSFORMATION, APPLIED_CLASSIFICATION_
    ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PROPERTY_DEFINITION := [];
    t2_set: SET OF ITEM_DEFINED_TRANSFORMATION := [];
    violation: BOOLEAN := FALSE;

```

```

END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'local co ordinate system');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                  'ITEM_DEFINED_TRANSFORMATION' + '.' + 'TRANSFORM_ITEM_1');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                i.NAME = 'transformation to parent')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.54 id_of_property_definition_with_class_local_co_ordinate_system_- matches_pattern

The id_of_property_definition_with_class_local_co_ordinate_system_matches_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. property_definition: T - type having attribute A; 'local co ordinate system': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_property_definition_with_class_local_co_ordinate_system_
  pattern FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'local co ordinate system') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:))) = 0;

```

END_RULE;

5.2.5.55 id_of_property_definition_with_class_spacing_table_matches_pattern

The id_of_property_definition_with_class_spacing_table_matches_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. property_definition: T - type having attribute A; 'spacing table': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_property_definition_with_class_spacing_table_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clified_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'spacing table') THEN
      clified_inst := clified_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clified_inst | NOT(i.id LIKE *))) = 0;
END_RULE;

```

5.2.5.56 PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_with_position_reference_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system_with_position_reference

The PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_with_position_reference_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system_with_position_reference gest all instances of T1 that have class id 'local co ordinate system with position reference' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "local co ordinate system with position reference" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_with_position_
reference_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_
DEFINITION_w_NAME_eq_local_co_ordinate_system_with_position_reference
FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'local co ordinate system with position reference');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'local co ordinate system with position reference')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.57 one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_ system_with_position_reference

The one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system_with_position_ref-
 erence forces the 'items' attribute of a representation to have for each entry in the list ('...')
 exactly one representation_item whose 'name' attribute has the value given in the list if the
 representation is the 'used_representation' in a property_definition_representation whose
 'name' attribute has a value VAL. 'local co ordinate system with position reference': VAL
 - value of the .name attribute of the property_definition_representation; 'local axes and
 origin': arg_list - list of names for the .name attributes of the representation_items that
 shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_local_co_ordinate_system_with_
    position_reference FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['local axes and origin'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'local co ordinate system with
                    position reference')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.58 one_rep_item_in_compound_rep_item_if_class_eq_spacing_position

The `one_rep_item_in_compound_rep_item_if_class_eq_spacing_position` forces the 'item_element' attribute of a `compound_representation_item` with the class id `CLASS` to have in the `list_representation_item` for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list. 'spacing position': `CLASS` - value of the class id of the `compound_representation_item`; 'position number', 'position': `arg_list` - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that `compound_representation_item`.

EXPRESS specification:

```

RULE one_rep_item_in_compound_rep_item_if_class_eq_spacing_position

```



```

FOR (compound_representation_item, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF COMPOUND_REPRESENTATION_ITEM := [];
  t2_set: LIST OF representation_item := [];
  arg_list: LIST OF STRING := ['position number', 'position'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'spacing position');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i:=1 TO HIINDEX(t1_set) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    t2_set := t1_set[i].item_element;
    violation := (SIZEOF(QUERY(items <* t2_set |
                               items.name = arg_list[j]))) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;

END_RULE;

```

5.2.5.59 one_rep_item_in_compound_rep_item_if_class_eq_spacing_position_-with_offset

The `one_rep_item_in_compound_rep_item_if_class_eq_spacing_position_with_offset` forces the 'item_element' attribute of a `compound_representation_item` with the class id `CLASS` to have in the `list_representation_item` for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list. 'spacing position with offset': `CLASS` - value of the class id of the `compound_representation_item`; 'offset': `arg_list` - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that `compound_representation_item`.

EXPRESS specification:

```

RULE one_rep_item_in_compound_rep_item_if_class_eq_spacing_position_with_offset
  FOR (compound_representation_item, APPLIED_CLASSIFICATION_ASSIGNMENT);

```

```

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF COMPOUND_REPRESENTATION_ITEM := [];
  t2_set: LIST OF representation_item := [];
  arg_list: LIST OF STRING := ['offset'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'spacing position with offset');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i:=1 TO HIINDEX(t1_set) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    t2_set := t1_set[i].item_element;
    violation := (SIZEOF(QUERY(items <* t2_set |
      items.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;

END_RULE;

```

5.2.5.60 one_compound_rep_item_with_class_spacing_position_in_compound_rep_item_with_class_spacing_position_with_offset

The one_compound_rep_item_with_class_spacing_position_in_compound_rep_item_with_class_spacing_position_with_offset forces the 'item_element' attribute of a compound_representation_item with the class id CLASS1 to have in the list_representation_item exactly one compound_representation_item with the class id CLASS2. 'spacing position with offset': CLASS1 - value of the class id of the compound_representation_item; 'spacing position': CLASS2 - value of the class id of the compound_representation_item, which should be referenced by the compound_representation_item with the class id CLASS1.

EXPRESS specification:

```

RULE one_compound_rep_item_with_class_spacing_position_in_compound_rep_item_
  with_class_spacing_position_with_offset FOR (compound_representation_item,

```

```

    APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
    c_a_set2 : SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
    t1_set:   SET OF COMPOUND_REPRESENTATION_ITEM := [];
    t2_set:   SET OF COMPOUND_REPRESENTATION_ITEM := [];
    t3_set:   SET OF REPRESENTATION_ITEM := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME =
    'spacing position with offset');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

c_a_set2 := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'spacing position');

REPEAT i := 1 TO HIINDEX(c_a_set2);
    REPEAT j := 1 TO HIINDEX(c_a_set2[i].items);
        t2_set := t2_set + c_a_set2[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i:=1 TO HIINDEX(t1_set) WHILE (NOT violation);
    REPEAT j := 1 TO HIINDEX(t1_set[i].item_element);
        t3_set := t3_set + t1_set[i].item_element[j];
    END_REPEAT;
    violation := (SIZEOF(t3_set * t2_set) <> 1);
    t3_set:= [];
END_REPEAT;

WHERE
    wr1: NOT violation;

END_RULE;

```

5.2.5.61 ID_of_product_definition_with_class_definable_object_pattern

The ID_of_product_definition_with_class_definable_object_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. product_definition: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; *.*: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_product_definition_with_class_definable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'definable object') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
  END_RULE;

```

5.2.5.62 ID_of_product_definition_relationship_with_class_definable_object_-pattern

The ID_of_product_definition_relationship_with_class_definable_object_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. product_definition_relationship: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_product_definition_relationship_with_class_definable_object_
  pattern FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'definable object') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
  END_RULE;

```

5.2.5.63 max_one_version_id_per_versionable_object

The `max_one_version_id_per_versionable_object` ensures that a type that is referenced by an `applied_identification_assignment` whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object FOR(APPLIED_IDENTIFICATION_
  ASSIGNMENT);

LOCAL
  version_ids:          SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  versionable_objects:  BAG OF GENERIC := [];
  duplicate:            BOOLEAN := FALSE;
END_LOCAL;

version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
  i.ROLE.NAME = 'version identifier');

REPEAT i := 1 TO HIINDEX(version_ids);
  versionable_objects := versionable_objects + version_ids[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
  REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    duplicate := versionable_objects[i] :=: versionable_objects[j];
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT duplicate;
END_RULE;

```

5.2.5.64 ID_of_extended_shape_aspect_with_class_definable_object_pattern

The `ID_of_extended_shape_aspect_with_class_definable_object_pattern` forces string type attribute A of type T to match the pattern P if the instance of T is of class C. `extended_shape_aspect`: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_extended_shape_aspect_with_class_definable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'definable object') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
END_RULE;

```

5.2.5.65 SHAPE_ASPECT_w_class_id_composite_feature_referenced_by_- exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_- DEFINITION_w_NAME_eq_composite_feature_design_parameters

The SHAPE_ASPECT_w_class_id_composite_feature_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_composite_feature_design_parameters gets all classification_assignment instances with id 'composite feature' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "composite feature design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_composite_feature_referenced_by_exactly_1_
  PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_composite_
  feature_design_parameters FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_
  REPRESENTATION, APPLIED_CLASSIFICATION_ASSIGNMENT);

  LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF SHAPE_ASPECT := [];
    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

```

```

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'composite feature');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                  'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                i.NAME = 'composite feature design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.66 ID_of_extended_shape_aspect_relationship_with_class_definable-object_pattern

The ID_of_extended_shape_aspect_relationship_with_class_definable_object_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. extended_shape_aspect_relationship: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_extended_shape_aspect_relationship_with_class_definable_object_
pattern FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name
      = 'definable object') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *))) = 0;
END_RULE;

```

5.2.5.67 ID_of_representation_with_class_versionable_object_pattern

The ID_of_representation_with_class_versionable_object_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. representation: T - type having attribute A; 'versionable object': C - class of T; ID: A - attribute of type T; *.*: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_representation_with_class_versionable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'versionable object') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *.:))) = 0;
END_RULE;

```

5.2.5.68 explicit_feature_design_definition_has_max_items_elements_in_N

The explicit_feature_design_definition_has_max_items_elements_in_N forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE explicit_feature_design_definition_has_max_items_elements_in_N
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(

```



```

        temp1 <* representation |
            SIZEOF (
                QUERY(
                    temp2 <* USEDIN(temp1,
'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
| ((temp2\classification_assignment.role.name = 'class membership')
AND (temp_appl_class\classification_assignment.
assigned_classification\group.name =
'explicit feature design definition')))) > 0 );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.N) > items)) = 0;
END_RULE;

```

5.2.5.69 comp_rep_item_w_name_eq_representations

The `comp_rep_item_w_name_eq_representations` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'representations': VAL - value of the inherited name attribute of the `compound_representation_item`; 'explicit feature design definition': `arg_list` - list of names for the .name attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_representations FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'explicit feature design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
    temp_comp_rep_item.name = 'explicit feature design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.70 one_item_in_rep_if_used_in_pdr_w_name_eq_maximum_permissible_SWSF_values

The `one_item_in_rep_if_used_in_pdr_w_name_eq_maximum_permissible_SWSF_values` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'name_eq_maximum_permissible_SWSF_values'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_maximum_permissible_SWSF_values
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := 'negative value', 'positive value',
                                'position';

    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name =
                'maximum permissible SWSF values')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.71 ID_of_extended_product_definition_shape_with_class_versionable_-object_pattern

The ID_of_extended_product_definition_shape_with_class_versionable_object_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. extended_product_definition_shape: T - type having attribute A; 'versionable object': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_extended_product_definition_shape_with_class_versionable_object_
  pattern FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
    'versionable object') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *:))) = 0;
END_RULE;

```

5.2.5.72 same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_assembly_relationship

The same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_assembly_relationship forces the relating and the related side of a relationship of type T_relationship to be of the same class if T_relationship is of class C. PRODUCT_DEFINITION: T - type that is related by T_relationship; 'assembly relationship': C - class of T_relationship.

EXPRESS specification:

```

RULE same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_
  assembly_relationship FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
  cl_ass:      SET OF CLASSIFICATION_ASSIGNMENT := [];
  clfied_inst: SET OF PRODUCT_DEFINITION_RELATIONSHIP := [];
END_LOCAL;

```

```

    find all instances that are of class 'assembly relationship'
cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'assembly relationship');

REPEAT i := 1 TO HIINDEX(cl_ass);
    clified_inst := clified_inst + cl_ass[i].items;
END_REPEAT;

WHERE
    WR1: SIZEOF(WHICH_CLASS(i.RELATING_PRODUCT_DEFINITION) - WHICH_CLASS(i.RELATED_PRODUCT_DEFINITION))
END_RULE;

```

5.2.5.73 product_definition_w_class_id_principal_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition

The product_definition_w_class_id_principal_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition forces that a product_definition with a class id C is referenced by one property_definition_representation with the name C via a property_definition. 'principal characteristics': C - class.name and property_definition_representation.name.

EXPRESS specification:

```

RULE product_definition_w_class_id_principal_characteristics_referenced_by_
    exactly_1_property_definition_representation_via_property_definition
    FOR (product_definition, property_definition, property_definition_
        representation, applied_classification_assignment);

LOCAL
    c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF product_definition := [];
    t2_set: SET OF property_definition_representation := [];
    t3_set: SET OF property_definition := [];
    t4_set: SET OF product_definition := [];

    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'principal characteristics');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;

```

```

END_REPEAT;

t2_set:= QUERY(i <* PROPERTY_DEFINITION_REPRESENTATION |
               i.NAME = 'principal characteristics');

REPEAT i := 1 TO HIINDEX(t2_set);
    t3_set := t3_set + t2_set[i].definition;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t3_set);
    t4_set := t4_set + t3_set[i].definition;
END_REPEAT;

violation := t1_set <> t4_set;

WHERE
    wr1: NOT violation;

END_RULE;

```

5.2.5.74 one_item_in_rep_if_used_in_pdr_w_name_eq_principal_characteristics

The `one_item_in_rep_if_used_in_pdr_w_name_eq_principal_characteristics` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value. 'principal characteristics': VAL - value of the .name attribute of the property_definition_representation; 'length between perpendiculars', 'moulded breadth', 'moulded depth', 'design draught', 'design deadweight', 'min daught at fp', 'max daught at fp', 'min daught at ap', 'max daught at ap': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_principal_characteristics
    FOR (representation);

    LOCAL
        reps:      BAG OF REPRESENTATION := [];
        arg_list:  LIST OF STRING := ['length between perpendiculars',
                                     'moulded breadth', 'moulded depth', 'design draught',
                                     'design deadweight', 'min daught at fp', 'max daught at fp',
                                     'min daught at ap', 'max daught at ap'];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    reps := QUERY(

```

```

        temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'principal characteristics')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.75 optional_occurrence_of_attributes_for_principal_characteristics

The `optional_occurrence_of_attributes_for_principal_characteristics` forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'principal characteristics': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'block coefficient': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_principal_characteristics
    FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['block coefficient'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(

```

```

                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'principal characteristics')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.76 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
    FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

LOCAL
    version_ids:          SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects:  BAG OF GENERIC := [];
    duplicate:            BOOLEAN := FALSE;
END_LOCAL;

version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
    i.ROLE.NAME = 'version identifier');

REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
        duplicate := versionable_objects[i] :=: versionable_objects[j];
    END_REPEAT;
END_REPEAT;

```

```

WHERE
    wr1: NOT duplicate;
END_RULE;

```

5.2.5.77 id_of_product_definition_with_class_principal_characteristics_pattern

The `id_of_product_definition_with_class_principal_characteristics_pattern` forces string type attribute A of type T to match the pattern P if the instance of T is of class C. product_definition: T - type having attribute A; 'principal characteristics': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_definition_with_class_principal_characteristics_
    pattern FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
    clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
        'principal characteristics') THEN
        clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
END_REPEAT;

WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:))) = 0;
END_RULE;

```

5.2.5.78 id_of_product_definition_with_class_class_and_statutory_- designations_pattern

The `id_of_product_definition_with_class_class_and_statutory_designations_pattern` forces string type attribute A of type T to match the pattern P if the instance of T is of class C. product_definition: T - type having attribute A; 'class and statutory designation': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_definition_with_class_class_and_statutory_designation_
    pattern FOR (CLASSIFICATION_ASSIGNMENT);

```



```

LOCAL
  clfied_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
  IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
    'class and statutory designation') THEN
    clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
  END_IF;
END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:))) = 0;
END_RULE;

```

5.2.5.79 id_of_product_definition_with_class_ship_designation_pattern

The `id_of_product_definition_with_class_ship_designation_pattern` forces string type attribute A of type T to match the pattern P if the instance of T is of class C. `product_definition`: T - type having attribute A; 'ship designation': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_definition_with_class_ship_designation_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'ship designation') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:))) = 0;
  END_RULE;

```

5.2.5.80 id_of_product_definition_with_class_shipyard_designation_pattern

The `id_of_product_definition_with_class_shipyard_designation_pattern` forces string type at-

tribute A of type T to match the pattern P if the instance of T is of class C. product_definition: T - type having attribute A; 'shipyard designation': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_definition_with_class_shipyard_designation_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name = '
      shipyard designation') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:))) = 0;
END_RULE;

```

5.2.5.81 id_of_product_definition_with_class_owner_designation_pattern

The id_of_product_definition_with_class_owner_designation_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. product_definition: T - type having attribute A; 'owner designation': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_definition_with_class_owner_designation_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'owner designation') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

```

```

END_REPEAT;

WHERE
  WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:)) = 0;
END_RULE;

```

5.2.5.82 id_of_product_related_product_category_with_class_ship_type-pattern

The id_of_product_related_product_category_with_class_ship_type_pattern forces string type attribute The of type T to match the pattern P if the instance of T is of class C. product_related_product_category: T - type having attribute A; 'ship type': C - class of T; id: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE id_of_product_related_product_category_with_class_ship_type_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
      'ship type') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.id LIKE *:)) = 0;
  END_RULE;

```

5.2.5.83 product_definition_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_property_definition_representation_via_property_definition

The product_definition_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_property_definition_representation_via_property_definition forces that a product_definition with a class id C is referenced by one property_definition_representation with the name C via a property_definition. 'class and statutory designation': C - class.name and property_definition_representation.name.

EXPRESS specification:

```

RULE product_definition_w_class_id_class_and_statutory_designation_referenced_
  by_exactly_1_property_definition_representation_via_property_definition
  FOR (product_definition, property_definition,
       property_definition_representation, applied_classification_assignment);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF product_definition := [];
  t2_set:  SET OF property_definition_representation := [];
  t3_set:  SET OF property_definition := [];
  t4_set:  SET OF product_definition := [];

  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME =
                 'class and statutory designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

t2_set:= QUERY(i <* PROPERTY_DEFINITION_REPRESENTATION |
               i.NAME = 'class and statutory designation');

REPEAT i := 1 TO HIINDEX(t2_set);
  t3_set := t3_set + t2_set[i].definition;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t3_set);
  t4_set := t4_set + t3_set[i].definition;
END_REPEAT;

violation := t1_set <> t4_set;

WHERE
  wr1: NOT violation;

END_RULE;

```

5.2.5.84 one_item_in_rep_if_used_in_pdr_w_name_eq_class_and_statutory_- designation

The one_item_in_rep_if_used_in_pdr_w_name_eq_class_and_statutory_designation forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'class and statutory designation': VAL - value of the .name attribute of the property_definition_representation; 'class number': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_class_and_statutory_designation
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['class number'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name =
          'class and statutory designation')
        )
      ) > 0
    );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;
```

5.2.5.85 PRODUCT_DEFINITION_w_class_id_class_and_statutory_- designated_referenced_by_exactly_1_PROPERTY_DEFINITION_- w_class_id_class_notation_via_DEFINITION

The PRODUCT_DEFINITION_w_class_id_class_and_statutory_designated_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_class_notation_via_DEFINITION.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_class_and_statutory_designated_referenced_
  by_exactly_1_PROPERTY_DEFINITION_w_class_id_class_notation_via_DEFINITION
  FOR (PRODUCT_DEFINITION, PROPERTY_DEFINITION, APPLIED_CLASSIFICATION_
  ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'class and statutory designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION'
  + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
  'class notation' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.86 PRODUCT_DEFINITION_w_class_id_class_and_statutory_- designated_referenced_by_exactly_1_PROPERTY_DEFINITION_- w_class_id_regulation_via_DEFINITION

The PRODUCT_DEFINITION_w_class_id_class_and_statutory_designated_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_regulation_via_DEFINITION.

exactly_1_PROPERTY_DEFINITION_w_class_id_regulation_via_DEFINITION specifies that all instances of PRODUCT_DEFINITION in t1_set: get the PROPERTY_DEFINITION instances that are referencing a PRODUCT_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION instances whose class id is 'regulation' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_class_and_statutory_designation_referenced_
  by_exactly_1_PROPERTY_DEFINITION_w_class_id_regulation_via_DEFINITION
  FOR (PRODUCT_DEFINITION, PROPERTY_DEFINITION, APPLIED_CLASSIFICATION_
  ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION := [];
  t2_set:  SET OF PROPERTY_DEFINITION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'class and statutory designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION' +
  '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
  'regulation' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.87 PROPERTY_DEFINITION_w_class_id_class_notation_referenced_- by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_- via_DEFINITION_w_NAME_eq_class_notation

The PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_class_notation

specifies that all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "class notation" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_class_notation
FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PROPERTY_DEFINITION := [];
  t2_set:  SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'class notation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                  'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                i.NAME = 'class notation')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.88 PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_class_society

The PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_class_society gets all classification_assignment instances with id 'class notation' for all instances of PROPERTY_

DEFINITION in t1_set: get the APPLIED_ORGANIZATION_ASSIGNMENT instances that are referencing a PROPERTY_DEFINITION instance via ITEMS, filter out those APPLIED_ORGANIZATION_ASSIGNMENT instances whose attribute ROLE has the value "class society" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_
  APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_class_society
  FOR(PROPERTY_DEFINITION, APPLIED_ORGANIZATION_ASSIGNMENT, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PROPERTY_DEFINITION := [];
  t2_set:  SET OF APPLIED_ORGANIZATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'class notation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE = 'class society')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.89 one_or_more_items_in_rep_if_used_in_pdr_w_name_eq_class_notation

The one_or_more_items_in_rep_if_used_in_pdr_w_name_eq_class_notation forces the 'items' attribute of a representation to have for each entry in the list ('...') one or more representation_items whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'class notation': VAL - value of the .name attribute of the representation; 'class notations hull', 'class notations machinery': arg_list - list of names for the .name attributes of the representation_items that shall belong one or more to that representation

EXPRESS specification:

```

RULE one_or_more_items_in_rep_if_used_in_pdr_w_name_eq_class_notation
  FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['class notations hull',
                                'class notations machinery'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'class notation')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) < 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.90 optional_occurrence_of_attributes_for_class_notation

The `optional_occurrence_of_attributes_for_class_notation` forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one `representation_item` who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` who's 'name' attribute has a value VAL. 'class notation': VAL - value of the .name attribute of the `property_definition_representation` that uses the representation; 'ice class notation', 'service factor' approval required for heavy cargo': arg_list - list of names for the .name attributes of the `representation_items` that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_class_notation FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['ice class notation', 'service factor',
                                'approval required for heavy cargo'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name = 'class notation')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.91 one_item_in_rep_if_used_in_pdr_w_name_eq_class_notation

The `one_item_in_rep_if_used_in_pdr_w_name_eq_class_notation` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation-item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'class notation': VAL - value of the .name attribute of the property_definition_representation; 'service area', 'approval required for oil cargo', 'approval required for loading unloading aground', 'approval required for unloading grabs': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_class_notation
  FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['service area',
                                'approval required for oil cargo', 'approval
                                required for loading unloading aground',
                                'approval required for unloading grabs'];

  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'class notation')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.92 product_definition_w_class_id_class_parameters_referenced_by_- exactly_1_property_definition_representation_via_property_- definition

The product_definition_w_class_id_class_parameters_referenced_by_exactly_1_property_definition_representation_via_property_definition forces that a product_definition with a class id C is referenced by one property_definition_representation with the name C via a property_definition. 'class parameters': C - class.name and property_definition_representation.name

EXPRESS specification:

```

RULE product_definition_w_class_id_class_parameters_referenced_by_exactly_1_
  property_definition_representation_via_property_definition
  FOR (product_definition, property_definition, property_definition_
  representation, applied_classification_assignment);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF product_definition := [];
  t2_set: SET OF property_definition_representation := [];
  t3_set: SET OF property_definition := [];
  t4_set: SET OF product_definition := [];

  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'class parameters');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

t2_set:= QUERY(i <* PROPERTY_DEFINITION_REPRESENTATION |
  i.NAME = 'class parameters');

REPEAT i := 1 TO HIINDEX(t2_set);
  t3_set := t3_set + t2_set[i].definition;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t3_set);
  t4_set := t4_set + t3_set[i].definition;
END_REPEAT;

violation := t1_set <> t4_set;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.93 one_item_in_rep_if_used_in_pdr_w_name_eq_class_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_class_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used-representation' in a property_definition_representation whose 'name' attribute has a value

VAL. 'class parameters': VAL - value of the .name attribute of the property_definition_-representation: 'length class', 'length solas', 'scantlings draught', 'block coefficient class', 'design speed ahead', 'design speed astern': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_class_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['length class', 'length solas',
                                'scantlings draught',
                                'block coefficient class',
                                'design speed ahead', 'design speed astern'];

    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'class parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.94 PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE

The PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE for all instances of PRODUCT_

DEFINITION in t1_set: get the APPLIED_IDENTIFICATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_IDENTIFICATION_ASSIGNMENT instances whose attribute ROLE.NAME has the value "IMO number" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_
  APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE.NAME_eq_IMO_number_or_
  pennant_hull_number
  FOR(PRODUCT_DEFINITION, APPLIED_IDENTIFICATION_ASSIGNMENT,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'ship designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_IDENTIFICATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | (i.ROLE.NAME = 'IMO number')
    OR (i.ROLE.NAME = 'pennant hull number') )) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.95 PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_call_sign

The PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_call_sign gets all classification_assignment instances with id 'ship designation' gets all instances of T1 that have

class id 'ship designation' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_IDENTIFICATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_IDENTIFICATION_ASSIGNMENT instances whose attribute ROLE has the value "call sign" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_
  APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_call_sign
  FOR(PRODUCT_DEFINITION, APPLIED_IDENTIFICATION_ASSIGNMENT,
    APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'ship designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_IDENTIFICATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE = 'call sign')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.96 PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_flag_state

The PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_flag_state gets all instances of T1 that have class id 'ship designation' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_IDENTIFICATION_ASSIGNMENT instances that are ref-

erencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_IDENTIFICATION_ASSIGNMENT instances whose attribute ROLE has the value "flag state" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_
  APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_flag_state
  FOR(PRODUCT_DEFINITION, APPLIED_IDENTIFICATION_ASSIGNMENT, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'ship designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_IDENTIFICATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE = 'flag state')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.97 PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_port_of_registration

The PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_port_of_registration gets all classification_assignment instances with id 'ship designation' for all instances of PRODUCT_DEFINITION in t1_set: gets the APPLIED_IDENTIFICATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_IDENTIFICATION_ASSIGNMENT instances whose attribute ROLE

has the value "port of registration" check if their number equals 1.

EXPRESS specification:

```
RULE PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_
  APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_port_of_registration
  FOR(PRODUCT_DEFINITION, APPLIED_IDENTIFICATION_ASSIGNMENT, APPLIED_
  CLASSIFICATION_ASSIGNMENT);
```

```
LOCAL
```

```
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION := [];
  t2_set:  SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
```

```
END_LOCAL;
```

```
c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'ship designation');
```

```
REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;
```

```
REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'APPLIED_IDENTIFICATION_ASSIGNMENT' + '.' + 'ITEMS
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE = 'port of registration')) = 1);
END_REPEAT;
```

```
WHERE
```

```
  wr1: NOT violation;
```

```
END_RULE;
```

5.2.5.98 PRODUCT_DEFINITION_w_class_id_shipyard_designation_- referenced_by_exactly_1_APPLIED_ORGANIZATION_- ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard

The PRODUCT_DEFINITION_w_class_id_shipyard_designation_referenced_by_exactly_1-APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard gets all classification_assignment instances with id 'shipyard designation' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_ORGANIZATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_ORGANIZATION_ASSIGNMENT instances whose attribute ROLE has the value "shipyard" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_shipyard_designation_referenced_by_exactly_
1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard
FOR(PRODUCT_DEFINITION, APPLIED_ORGANIZATION_ASSIGNMENT, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION := [];
  t2_set:  SET OF APPLIED_ORGANIZATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'shipyard designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                  'APPLIED_ORGANIZATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE = 'shipyard')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.99 PRODUCT_DEFINITION_w_class_id_owner_designation_- referenced_by_exactly_1_APPLIED_ORGANIZATION_- ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company

The PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company gets all instances of T1 that have class id 'owner designation' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_ORGANIZATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_ORGANIZATION_ASSIGNMENT instances whose attribute ROLE has the value "managing company" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_
  APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company
  FOR(PRODUCT_DEFINITION, APPLIED_ORGANIZATION_ASSIGNMENT, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_ORGANIZATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'owner designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.ROLE = 'managing company')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.100 **PRODUCT_DEFINITION_w_class_id_owner_designation_- referenced_by_exactly_1_APPLIED_ORGANIZATION_- ASSIGNMENT_via_ITEMS_w_ROLE_eq_ordering_company**

The PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_ordering_company gets all instances of T1 that have class id 'owner designation' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_ORGANIZATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_ORGANIZATION_ASSIGNMENT instances whose attribute ROLE has the value "ordering company" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_
  APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_ordering_company
  FOR(PRODUCT_DEFINITION, APPLIED_ORGANIZATION_ASSIGNMENT, APPLIED_
    CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_ORGANIZATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'owner designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.ROLE = 'ordering company')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.101 **PRODUCT_DEFINITION_w_class_id_owner_designation_- referenced_by_exactly_1_APPLIED_ORGANIZATION_- ASSIGNMENT_via_ITEMS_w_ROLE_eq_owning_company**

The PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_owning_company gets all classification_assignment instances with id 'owner designation' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_ORGANIZATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_ORGANIZATION_ASSIGNMENT instances whose attribute ROLE has the value "owning company" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_
  APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_owning_company
  FOR(PRODUCT_DEFINITION, APPLIED_ORGANIZATION_ASSIGNMENT, APPLIED_
    CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_ORGANIZATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'owner designation');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.ROLE = 'owning company')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.102 product_definition_w_class_id_freeboard_characteristics_- referenced_by_exactly_1_property_definition_representation_via_- property_definition

The product_definition_w_class_id_freeboard_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition forces that a product_definition with a class id C is referenced by one property_definition_representation with the name C via a property_definition 'freeboard characteristics': C - class.name and property_definition_representation.name.

EXPRESS specification:

```

RULE product_definition_w_class_id_freeboard_characteristics_referenced_by_

```

```

exactly_1_property_definition_representation_via_property_definition
FOR (product_definition, property_definition, property_definition_
representation, applied_classification_assignment);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF product_definition := [];
  t2_set:  SET OF property_definition_representation := [];
  t3_set:  SET OF property_definition := [];
  t4_set:  SET OF product_definition := [];

  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME =
                 'freeboard characteristics');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

t2_set:= QUERY(i <* PROPERTY_DEFINITION_REPRESENTATION |
              i.NAME = 'freeboard characteristics');

REPEAT i := 1 TO HIINDEX(t2_set);
  t3_set := t3_set + t2_set[i].definition;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t3_set);
  t4_set := t4_set + t3_set[i].definition;
END_REPEAT;

violation := t1_set <> t4_set;

WHERE
  wr1: NOT violation;

END_RULE;

```

5.2.5.103 PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_-referenced_by_exactly_1_APPLIED_DATE_AND_TIME_ASSIGNMENT_via_ITEMS_w_ROLE_eq_date_freeboard_assigned

The PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_DATE_AND_TIME_ASSIGNMENT_via_ITEMS_w_ROLE_eq_date_freeboard_assigned gets all classification_assignment instances with id 'freeboard characteristics' for all instances of PRODUCT_DEFINITION in t1_set: get the APPLIED_DATE_AND-

TIME_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_DATE_AND_TIME_ASSIGNMENT instances whose attribute ROLE has the value "date freeboard assigned" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_
  exactly_1_APPLIED_DATE_AND_TIME_ASSIGNMENT_via_ITEMS_w_ROLE_eq_date_
  freeboard_assigned
  FOR(PRODUCT_DEFINITION, APPLIED_DATE_AND_TIME_ASSIGNMENT, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_DATE_AND_TIME_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'freeboard characteristics');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE =
    'date freeboard assigned')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.104 PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_-referenced_by_exactly_1_APPLIED_ORGANIZATION_-ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_assigned_by

The PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_assigned_by gets all instances of T1 that have class id 'freeboard characteristics' for all instances

of PRODUCT_DEFINITION in t1_set: get the APPLIED_ORGANIZATION_ASSIGNMENT instances that are referencing a PRODUCT_DEFINITION instance via ITEMS, filter out those APPLIED_ORGANIZATION_ASSIGNMENT instances whose attribute ROLE has the value "freeboard assigned by" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_
  exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_
  assigned_by
  FOR(PRODUCT_DEFINITION, APPLIED_ORGANIZATION_ASSIGNMENT, APPLIED_
    CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION := [];
  t2_set: SET OF APPLIED_ORGANIZATION_ASSIGNMENT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'freeboard characteristics');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE =
    'freeboard assigned by')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.105 one_item_in_rep_if_used_in_pdr_w_name_eq_freeboard_ **characteristics**

The one_item_in_rep_if_used_in_pdr_w_name_eq_freeboard_characteristics forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has

a value VAL. 'freeboard characteristics': VAL - value of the .name attribute of the property_definition_representation; 'freeboard': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_freeboard_characteristics
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['freeboard'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'freeboard characteristics')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.106 PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_-referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_loadline_via_DEFINITION

The PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_PROPERTY_DEFINITION_w_class_id_loadline_via_DEFINITION gets all classification_assignment instances with id 'freeboard characteristics' for all instances of PRODUCT_DEFINITION in t1_set: get the PROPERTY_DEFINITION instances that are referencing a PRODUCT_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION instances whose class id is 'loadline' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_
  exactly_1_PROPERTY_DEFINITION_w_class_id_loadline_via_DEFINITION
  FOR(PRODUCT_DEFINITION, PROPERTY_DEFINITION, APPLIED_CLASSIFICATION_
  ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION := [];
  t2_set:  SET OF PROPERTY_DEFINITION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'freeboard characteristics');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION' +
  '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
  'loadline' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.107 **PROPERTY_DEFINITION_w_class_id_loadline_referenced_by_- exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_- DEFINITION_w_NAME_eq_loadline**

The PROPERTY_DEFINITION_w_class_id_loadline_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_loadline gets all classification_assignment instances with id 'loadline' gets all instances of T1 that have class id 'loadline' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "loadline" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_loadline_referenced_by_exactly_1_
  PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_loadline
  FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'loadline');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'loadline')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.108 one_item_in_rep_if_used_in_pdr_w_name_eq_loadline

The `one_item_in_rep_if_used_in_pdr_w_name_eq_loadline` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used-representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'loadline': VAL - value of the .name attribute of the property_definition_representation; 'load line length', 'load line depth', 'load line displacement', 'load line block coefficient', 'load line regulation': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_loadline FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['load line length', 'load line depth',
                                'load line displacement',
                                'load line block coefficient',
                                'load line regulation'];

  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'loadline')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.109 product_definition_w_class_id_lightship_definition_referenced_by_exactly_1_property_definition_representation_via_property_definition

The product_definition_w_class_id_lightship_definition_referenced_by_exactly_1_property_definition_representation_via_property_definition forces that a product_definition with a class id C is referenced by one property_definition_representation with the name C via a property_definition 'lightship_definition': C - class.name and property_definition_representation.name.

EXPRESS specification:

```

RULE product_definition_w_class_id_lightship_definition_referenced_by_exactly_
  1_property_definition_representation_via_property_definition
  FOR (product_definition, property_definition, property_definition_
    representation, applied_classification_assignment);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF product_definition := [];
  t2_set: SET OF property_definition_representation := [];
  t3_set: SET OF property_definition := [];
  t4_set: SET OF product_definition := [];

  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'lightship_definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

t2_set:= QUERY(i <* PROPERTY_DEFINITION_REPRESENTATION |
  i.NAME = 'lightship_definition');

REPEAT i := 1 TO HIINDEX(t2_set);
  t3_set := t3_set + t2_set[i].definition;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t3_set);
  t4_set := t4_set + t3_set[i].definition;
END_REPEAT;

violation := t1_set <> t4_set;

WHERE
  wr1: NOT violation;

END_RULE;

```

5.2.5.110 one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_definition

The `one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_definition` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the

'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'lightship_definition': VAL - value of the .name attribute of the property_definition_representation; 'lightship weight', 'lightship centre of gravity': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_definition
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['lightship weight',
                                  'lightship centre of gravity'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'lightship_definition')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.111 PROPERTY_DEFINITION_w_class_id_lightship_weight_item_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_lightship_weight_item

The PROPERTY_DEFINITION_w_class_id_lightship_weight_item_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_light-

ship_weight_item gets all classification_assignment instances with id 'lightship weight item' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "lightship weight item" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_lightship_weight_item_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  lightship_weight_item
  FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PROPERTY_DEFINITION := [];
  t2_set:  SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'lightship weight item');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'lightship weight item')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.112 one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_weight_item

The one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_weight_item forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the

'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'lightship weight item': VAL - value of the .name attribute of the property_definition_representation; 'aft weight extent', 'fwd weight extent': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_lightship_weight_item
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['aft weight extent', 'fwd weight extent'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'lightship weight item')
        )
      ) > 0
    );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.113 PROPERTY_DEFINITION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weight_and_centre_of_gravity

The PROPERTY_DEFINITION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weight_and_centre_of_gravity gets all classification_assignment instances with id 'weight

and centre of gravity' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "weight and centre of gravity" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_weight_and_centre_of_gravity_referenced_
  by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  weight_and_centre_of_gravity
  FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'weight and centre of gravity');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'weight and centre of gravity')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.114 one_item_in_rep_if_used_in_pdr_w_name_eq_weight_and_centre_of_gravity

The one_item_in_rep_if_used_in_pdr_w_name_eq_weight_and_centre_of_gravity forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is

the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'weight and centre of gravity': VAL - value of the .name attribute of the property_definition_representation; 'mass ', 'centre of gravity': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_weight_and_centre_of_gravity
  FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['mass ', 'centre of gravity'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'weight and centre of gravity')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.115 PROPERTY_DEFINITION_REPRESENTATION_w_class_id_- weight_and_centre_of_gravity_referenced_by_exactly_1_- REPRESENTATION_via_USED_REPRESENTATION_w_- NAME_eq_moment_3d

The PROPERTY_DEFINITION_REPRESENTATION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_REPRESENTATION_via_USED_REPRESENTATION_w_NAME_eq_moment_3d for all instances of PROPERTY_DEFINITION_REPRESENTA-

TION in t1_set: get the REPRESENTATION instances that are referencing a PROPERTY_DEFINITION_REPRESENTATION instance via USED_REPRESENTATION, filter out those REPRESENTATION instances whose attribute NAME has the value "moment 3d" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_REPRESENTATION_w_class_id_weight_and_centre_of_
gravity_referenced_by_exactly_1_REPRESENTATION_via_USED_REPRESENTATION_w_
NAME_eq_moment_3d
FOR(PROPERTY_DEFINITION_REPRESENTATION, REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  t2_set: SET OF REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'weight and centre of gravity');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'REPRESENTATION' +
    '.' + 'USED_REPRESENTATION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'moment 3d')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.116 one_item_in_rep_if_name_eq_moment_3d

The one_item_in_rep_if_name_eq_moment_3d forces the 'items' attribute of a representation with the name VAL to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list 'moment 3d': VAL - value of the .name attribute of the representation; 'longitudinal moment', 'transverse moment', 'vertical moment', 'origin': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_name_eq_moment_3d FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['longitudinal moment', 'transverse moment',
                                'vertical moment', 'origin'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(i <* representation |
              i.NAME = 'moment 3d');

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.117 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object FOR(APPLIED_IDENTIFICATION_
ASSIGNMENT);

LOCAL
    version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects: BAG OF GENERIC := [];
    duplicate:        BOOLEAN := FALSE;
END_LOCAL;

version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
                    i.ROLE.NAME = 'version identifier');

REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;

```

```

END_REPEAT;

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
  REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    duplicate := versionable_objects[i] :=: versionable_objects[j];
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT duplicate;
END_RULE;

```

5.2.5.118 **PROPERTY_DEFINITION_w_class_id_ship_material_property_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_ship_-material_property_parameters**

The PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_ship_material_property_parameters gets all classification_assignment instances with id 'ship material property' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "ship material property parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_ship_
  material_property_parameters
  FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'ship material property');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];

```

```

    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PROPERTY_DEFINITION_
        REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
        i.NAME = 'ship material property parameters')) = 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.119 one_item_in_rep_if_used_in_pdr_w_name_eq_ship_material_ property_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_ship_material_property_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'ship material property parameters': VAL - value of the .name attribute of the property_definition_representation; 'density': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_ship_material_property_parameters
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['density'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'ship material property parameters')
            )
        ) > 0
    );

```

```

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.120 PROPERTY_DEFINITION_w_class_id_ship_material_property_-referenced_by_exactly_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE

The PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE gets all classification_assignment instances with id 'ship material property' for all instances of PROPERTY_DEFINITION in t1_set: get the APPLIED_DOCUMENT_REFERENCE instances that are referencing a PROPERTY_DEFINITION instance via ITEMS, filter out those APPLIED_DOCUMENT_REFERENCE instances whose attribute ROLE.NAME has the value "material reference" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_
  exactly_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE.NAME_eq_material_
  reference
  FOR(PROPERTY_DEFINITION, APPLIED_DOCUMENT_REFERENCE, APPLIED_CLASSIFICATION_
  ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF APPLIED_DOCUMENT_REFERENCE := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'ship material property');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

```



```

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'APPLIED_DOCUMENT_REFERENCE' + '.' + 'ITEMS');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.ROLE.NAME = 'material reference')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.121 **PROPERTY_DEFINITION_w_class_id_homogeneous_ship_- material_property_referenced_by_exactly_1_PROPERTY_- DEFINITION_REPRESENTATION_via_DEFINITION_w_- NAME_eq_homogeneous_ship_material_property_parameters**

The PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_homogeneous_ship_material_property_parameters gets all classification_assignment instances with id 'homogeneous ship material property' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "homogeneous ship material property parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_homogeneous_ship_material_property_parameters
FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'homogeneous ship material property');

REPEAT i := 1 TO HIINDEX(c_a_set);

```

```

REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
        'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
        i.NAME = 'homogeneous ship material property parameters')) = 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.122 one_item_in_rep_if_used_in_pdr_w_name_eq_homogeneous_ship-material_property_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_homogeneous_ship-material_property_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'homogeneous ship material property parameters': VAL - value of the .name attribute of the property_definition_representation; 'density', 'poisson ratio', 'stress of fracture', 'thermal expansion coefficient', 'yield point', 'youngs module': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_homogeneous_ship_material_
    property_parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['density', 'poisson ratio',
        'stress of fracture',
        'thermal expansion coefficient',
        'yield point', 'youngs module'];

    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (

```

```

        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name =
                'homogeneous ship material property parameters')
        )
    ) > 0
);

iterate over all representations found above; stop, if one of
    them has not exactly one rep_item with for each name of the arg_list

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.123 PROPERTY_DEFINITION_w_class_id_homogeneous_ship_- material_property_referenced_by_at_most_1_APPLIED_- DOCUMENT_REFERENCE_via_ITEMS_w_ROLE

The PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_at_most_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE gets all classification_assignment instances with id 'homogeneous_ship material property' for all instances of PROPERTY_DEFINITION in t1_set: get the APPLIED_DOCUMENT_REFERENCE instances that are referencing a PROPERTY_DEFINITION instance via ITEMS, filter out those APPLIED_DOCUMENT_REFERENCE instances whose attribute ROLE.NAME has the value 'material reference' check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_
referenced_by_at_most_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE.NAME_
eq_material_reference
FOR(PROPERTY_DEFINITION, APPLIED_DOCUMENT_REFERENCE, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PROPERTY_DEFINITION := [];
    t2_set: SET OF APPLIED_DOCUMENT_REFERENCE := [];

```

```

    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'homogeneous_ship material property');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'APPLIED_DOCUMENT_REFERENCE' + '.' + 'ITEMS');
    violation := NOT (SIZEOF(QUERY(i <* t2_set | i.ROLE.NAME = 'material reference')) > 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.124 max_4_in_derived_unit_with_name_eq_angular_stiffness_unit_via- elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'angular stiffness unit': V11 - string value of attribute A11; 4: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements .

*)

EXPRESS specification:

```

RULE max_4_in_derived_unit_with_name_eq_angular_stiffness_unit_via_elements
    FOR (derived_unit);

    LOCAL
        t1_set: SET OF derived_unit := [];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    t1_set := QUERY(i <* derived_unit |
        i.name = 'angular stiffness unit');

```

```

REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 4;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.125 max_2_in_derived_unit_with_name_eq_density_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'density unit': V11 - string value of attribute A11; 2: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_2_in_derived_unit_with_name_eq_density_unit_via_elements
    FOR (derived_unit);

    LOCAL
        t1_set: SET OF derived_unit := [];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    t1_set := QUERY(i <* derived_unit |
                    i.name = 'density unit');

    REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
        violation := SIZEOF(t1_set.elements) > 2;
    END_REPEAT;

    WHERE
        wr1: NOT violation;
END_RULE;

```

5.2.5.126 max_2_in_derived_unit_with_name_eq_dilatation_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the

attribute that contains value V11; 'dilatation unit': V11 - string value of attribute A11; 2: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_2_in_derived_unit_with_name_eq_dilatation_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |
    i.name = 'dilatation unit');

  REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 2;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.127 max_2_in_derived_unit_with_name_eq_flow_rate_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V11; 'flow rate unit': V11 - string value of attribute A11; 2: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_2_in_derived_unit_with_name_eq_flow_rate_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

```

```

t1_set := QUERY(i <* derived_unit |
                i.name = 'flow rate unit');

REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 2;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.128 max_3_in_derived_unit_with_name_eq_force_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'force unit': V1 - string value of attribute A11; 3: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_3_in_derived_unit_with_name_eq_force_unit_via_elements
    FOR (derived_unit);

    LOCAL
        t1_set: SET OF derived_unit := [];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    t1_set := QUERY(i <* derived_unit |
                    i.name = 'force unit');

    REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
        violation := SIZEOF(t1_set.elements) > 3;
    END_REPEAT;

    WHERE
        wr1: NOT violation;
END_RULE;

```

5.2.5.129 max_1_in_derived_unit_with_name_eq_frequency_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V11; 'frequency unit': V11 - string value of attribute A11; 1: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_1_in_derived_unit_with_name_eq_frequency_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |
                  i.name = 'frequency unit');

  REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 1;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.130 max_3_in_derived_unit_with_name_eq_heat_rate_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V11; 'heat rate unit': V11 - string value of attribute A11; 3: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_3_in_derived_unit_with_name_eq_heat_rate_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

```



```

t1_set := QUERY(i <* derived_unit |
                i.name = 'heat rate unit');

REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 3;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.131 max_1_in_derived_unit_with_name_eq_inertia_moment_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'inertia moment unit': V11 - string value of attribute A11; 1: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

*)

EXPRESS specification:

```

RULE max_1_in_derived_unit_with_name_eq_inertia_moment_unit_via_elements
    FOR (derived_unit);

    LOCAL
        t1_set: SET OF derived_unit := [];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    t1_set := QUERY(i <* derived_unit |
                    i.name = 'inertia moment unit');

    REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
        violation := SIZEOF(t1_set.elements) > 1;
    END_REPEAT;

    WHERE
        wr1: NOT violation;
END_RULE;

```

5.2.5.132 max_3_in_derived_unit_with_name_eq_linear_stiffness_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'linear stiffness unit': V11 - string value of attribute A11; 3: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_3_in_derived_unit_with_name_eq_linear_stiffness_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |
    i.name = 'linear stiffness unit');

  REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 3;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.133 max_3_in_derived_unit_with_name_eq_pressure_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'pressure unit': V11 - string value of attribute A11; 3: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_3_in_derived_unit_with_name_eq_pressure_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |
                  i.name = 'pressure unit');

  REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 3;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.134 max_1_in_derived_unit_with_name_eq_product_of_inertia_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'product of inertia unit': V11 - string value of attribute A11; 1: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_1_in_derived_unit_with_name_eq_product_of_inertia_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |
                  i.name = 'product of inertia unit');

  REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 1;
  END_REPEAT;

```

```

END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.135 max_1_in_derived_unit_with_name_eq_section_modulus_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'section modulus unit': V11 - string value of attribute A11; 1: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_1_in_derived_unit_with_name_eq_section_modulus_unit_via_elements
    FOR (derived_unit);

    LOCAL
        t1_set: SET OF derived_unit := [];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    t1_set := QUERY(i <* derived_unit |
                    i.name = 'section modulus unit');

    REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
        violation := SIZEOF(t1_set.elements) > 1;
    END_REPEAT;

    WHERE
        wr1: NOT violation;
END_RULE;

```

5.2.5.136 max_2_in_derived_unit_with_name_eq_speed_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'speed unit': V11 - string value of attribute A11; 2:

N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_2_in_derived_unit_with_name_eq_speed_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |
    i.name = 'speed unit');

  REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 2;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.137 max_3_in_derived_unit_with_name_eq_stress_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V11; 'stress unit': V11 - string value of attribute A11; 3: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_3_in_derived_unit_with_name_eq_stress_unit_via_elements
  FOR (derived_unit);

  LOCAL
    t1_set: SET OF derived_unit := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  t1_set := QUERY(i <* derived_unit |

```

```

        i.name = 'stress unit');

REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
    violation := SIZEOF(t1_set.elements) > 3;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.138 max_4_in_derived_unit_with_name_eq_voltage_unit_via_elements

Each T1 with value V1 in attribute A11 shall have maximum N references to an aggregate via A12.

derived_unit: T1 - entity type of the referencing instance; name: A11 - name of the attribute that contains value V1; 'voltage unit': V11 - string value of attribute A11; 4: N - maximum number of T2 in T1.A12; elements: A12 - name of the attribute from T1 that references the aggregate that shall be maximum N elements.

EXPRESS specification:

```

RULE max_4_in_derived_unit_with_name_eq_voltage_unit_via_elements
    FOR (derived_unit);

    LOCAL
        t1_set: SET OF derived_unit := [];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    t1_set := QUERY(i <* derived_unit |
        i.name = 'voltage unit');

    REPEAT i := 1 TO HIINDEX(t1_set) WHILE NOT violation;
        violation := SIZEOF(t1_set.elements) > 4;
    END_REPEAT;

    WHERE
        wr1: NOT violation;
END_RULE;

```

5.2.5.139 uncertainty_measure_with_unit_has_description

The uncertainty_measure_with_unit_has_description forces the existence of an OPTIONAL attribute A for all instances of type T. uncertainty_measure_with_unit: T - type having attribute A; description: A - optional attribute who's existence shall be enforced.

EXPRESS specification:

```

RULE uncertainty_measure_with_unit_has_description
  FOR (uncertainty_measure_with_unit);

  WHERE
    WR1: SIZEOF(QUERY(i <* uncertainty_measure_with_unit |
      NOT(EXISTS (uncertainty_measure_with_unit.description)))) = 0;
END_RULE;

```

5.2.5.140 optional_occurrence_of_attributes_for_seam_curve_relationship_parameter

The optional_occurrence_of_attributes_for_seam_curve_relationship_parameter forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'seam curve relationship parameter': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_seam_curve_relationship_parameter FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := 'displacement';
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name =
          'seam curve relationship parameter')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);

```

```

        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
                               rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.141 seam_curve_relationship_has_max_items_elements_in_0

The `seam_curve_relationship_has_max_items_elements_in_0` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE seam\_curve_relationship_has_max_items_elements_in_0 FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
                | ((temp2\classification_assignment.role.name =
                    'class membership') AND
                    (temp_appl_class\classification_assignment.
                    assigned_classification\group.name = 'seam curve relationship'))
            )
        ) > 0
    );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
                     SIZEOF(temp3.0) > items)) = 0;
END_RULE;

```

5.2.5.142 max_one_version_id_per_versionable_object

The `max_one_version_id_per_versionable_object` ensures that a type that is referenced by an `applied_identification_assignment` whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
  FOR(APPLIED_IDENTIFICATION_ASSIGNMENT);

  LOCAL
    version_ids:          SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects:  BAG OF GENERIC := [];
    duplicate:            BOOLEAN := FALSE;
  END_LOCAL;

  version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
    i.ROLE.NAME = 'version identifier');

  REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
  END_REPEAT;

  REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
      duplicate := versionable_objects[i] :=: versionable_objects[j];
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT duplicate;
END_RULE;

```

5.2.5.143 bevel_design_definition_has_max_items_elements_in_3

The `bevel_design_definition_has_max_items_elements_in_3` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE bevel_design_definition_has_max_items_elements_in_3 FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
      QUERY(
        temp2 <* USEDIN(temp1,

```

```

'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
| ((temp2\classification_assignment.role.name =
    'class membership') AND
    (temp_appl_class\classification_assignment.
      assigned_classification\group.name
      = 'bevel design definition'))
    )
  ) > 0
);
WHERE
  wr1: SIZEOF(QUERY(temp3 <* var_representation |
    SIZEOF(temp3.3) > items)) = 0;
END_RULE;

```

5.2.5.144 comp_rep_item_w_name_eq_offset

The comp_rep_item_w_name_eq_offset forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'offset': VAL - value of the inherited name attribute of the compound_representation_item; 'bevel design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_offset FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'bevel design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name = 'bevel design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
      rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.145 comp_rep_item_w_name_eq_x_y_angle

The `comp_rep_item_w_name_eq_x_y_angle` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'x y angle': VAL - value of the inherited name attribute of the `compound_representation_item`; 'bevel design definition': `arg_list` - list of names for the .name attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_y_angle FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'bevel design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'bevel design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HIINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                         rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.146 circular_cutout_design_definition_has_max_items_elements_in_1

The `circular_cutout_design_definition_has_max_items_elements_in_1` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE circular_cutout_design_definition_has_max_items_elements_in_1
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
      QUERY(
        temp2 <* USEDIN(temp1,
          'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
        | ((temp2\classification_assignment.role.name =
          'class membership') AND
          (temp_appl_class\classification_assignment.
            assigned_classification\group.name
            = 'circular cutout design definition'))
      )
    ) > 0
  );

  WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
      SIZEOF(temp3.1) > items)) = 0;
END_RULE;

```

5.2.5.147 comp_rep_item_w_name_eq_radius

The comp_rep_item_w_name_eq_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'radius': VAL - value of the inherited name attribute of the compound_representation_item; 'circular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_radius FOR (compound_representation_item);

  LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'circular cutout design definition';
    violation:  BOOLEAN := FALSE;
  END_LOCAL;

```

```

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'circular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                rep_item(k).name = arg_list[j]))) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.148 drain_hole_cutout_design_definition_has_max_items_elements_in_5

The `drain_hole_cutout_design_definition_has_max_items_elements_in_5` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE drain_hole_cutout_design_definition_has_max_items_elements_in_5
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
      QUERY(
        temp2 <* USEDIN(temp1,
        'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
        | ((temp2\classification_assignment.role.name =
        'class membership') AND
        (temp_appl_class\classification_assignment.
        assigned_classification\group.name
        = 'drain hole cutout design definition'))
      )
    ) > 0
  );

  WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
                     SIZEOF(temp3.5) > items)) = 0;

```

END_RULE;

5.2.5.149 comp_rep_item_w_name_eq_depth

The comp_rep_item_w_name_eq_depth forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'depth': VAL - value of the inherited name attribute of the compound_representation_item; 'drain hole cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_depth FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'drain hole cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'drain hole cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j]))) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.150 comp_rep_item_w_name_eq_drain_hole_radius

The comp_rep_item_w_name_eq_drain_hole_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'drain hole radius': VAL - value of the inherited name attribute of the compound_representation_item; 'drain hole cutout design definition': arg_list - list

of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_drain_hole_radius
  FOR (compound_representation_item);

  LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'drain hole cutout design definition';
    violation:  BOOLEAN := FALSE;
  END_LOCAL;

  creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                  temp_comp_rep_item.name =
                    'drain hole cutout design definition');

  REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
      REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                      rep_item(k).name = arg_list[j]))) = 1);
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.151 comp_rep_item_w_name_eq_gap

The comp_rep_item_w_name_eq_gap forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'gap': VAL - value of the inherited name attribute of the compound_representation_item; 'drain hole cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_gap FOR (compound_representation_item);

  LOCAL

```

```

creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
arg_list:   LIST OF STRING := 'drain hole cutout design definition';
violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'drain hole cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.152 comp_rep_item_w_name_eq_gap_radius

The comp_rep_item_w_name_eq_gap_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'gap radius': VAL - value of the inherited name attribute of the compound_representation_item; 'drain hole cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_gap_radius FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'drain hole cutout design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'drain hole cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);

```



```

    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
            rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.153 comp_rep_item_w_name_eq_width

The comp_rep_item_w_name_eq_width forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'width': VAL - value of the inherited name attribute of the compound_representation_item; 'drain hole cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_width FOR (compound_representation_item);

LOCAL
    creps:    BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list: LIST OF STRING := 'drain hole cutout design definition';
    violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
    temp_comp_rep_item.name =
        'drain hole cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
            rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.154 edge_cutout_functional_definition_has_max_items_elements_in_1

The `edge_cutout_functional_definition_has_max_items_elements_in_1` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE edge_cutout_functional_definition_has_max_items_elements_in_1 FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
            | ((temp2\classification_assignment.role.name =
                'class membership') AND
                (temp_appl_class\classification_assignment.
                assigned_classification\group.name =
                'edge cutout functional definition'))
            )
        ) > 0
    );
WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.1) > items)) = 0;
END_RULE;

```

5.2.5.155 comp_rep_item_w_name_eq_the_function

The `comp_rep_item_w_name_eq_the_function` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'the function': VAL - value of the inherited name attribute of the `compound_representation_item`; 'edge cutout functional definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_the_function FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'edge cutout functional definition';
    violation:   BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'edge cutout functional definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                         rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.156 elliptical_cutout_design_definition_has_max_items_elements_in_2

The `elliptical_cutout_design_definition_has_max_items_elements_in_2` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE elliptical_cutout_design_definition_has_max_items_elements_in_2
FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
        QUERY(
            temp2 <* USEDIN(temp1,
                'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')

```

```

        | ((temp2\classification_assignment.role.name =
          'class membership') AND
          (temp_appl_class\classification_assignment.
            assigned_classification\group.name =
            'elliptical cutout design definition'))
        )
      ) > 0
    );
WHERE
  wr1: SIZEOF(QUERY(temp3 <* var_representation |
    SIZEOF(temp3.2) > items)) = 0;
END_RULE;

```

5.2.5.157 comp_rep_item_w_name_eq_half_axis_a

The comp_rep_item_w_name_eq_half_axis_a forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'half axis a': VAL - value of the inherited name attribute of the compound_representation_item; 'elliptical cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_half_axis_a FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'elliptical cutout design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name =
  'elliptical cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
      rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.158 comp_rep_item_w_name_eq_half_axis_b

The `comp_rep_item_w_name_eq_half_axis_b` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'half axis b': VAL - value of the inherited name attribute of the `compound_representation_item`; 'elliptical cutout design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_half_axis_b FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'elliptical cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'elliptical cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j]))) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.159 elongated_oval_cutout_design_definition_has_max_items_elements_in_6

The `elongated_oval_cutout_design_definition_has_max_items_elements_in_6` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to

have maximum N elements.

EXPRESS specification:

```

RULE elongated_oval_cutout_design_definition_has_max_items_elements_in_6
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
      QUERY(
        temp2 <* USEDIN(temp1,
          'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
        | ((temp2\classification_assignment.role.name =
          'class membership') AND
          (temp_appl_class\classification_assignment.
            assigned_classification\group.name =
            'elongated oval cutout design definition'))
        )
      ) > 0
    );

  WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
      SIZEOF(temp3.6) > items)) = 0;
END_RULE;

```

5.2.5.160 comp_rep_item_w_name_eq_distance

The comp_rep_item_w_name_eq_distance forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'distance': VAL - value of the inherited name attribute of the compound_representation_item; 'elongated oval cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_distance FOR (compound_representation_item);

  LOCAL

```

```

    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'elongated oval cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'elongated oval cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.161 comp_rep_item_w_name_eq_r1

The comp_rep_item_w_name_eq_r1 forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'r1': VAL - value of the inherited name attribute of the compound_representation_item; 'elongated oval cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_r1 FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'elongated oval cutout design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'elongated oval cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);

```

```

REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.162 comp_rep_item_w_name_eq_r2

The comp_rep_item_w_name_eq_r2 forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'r2': VAL - value of the inherited name attribute of the compound_representation_item; 'elongated oval cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_r2 FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'elongated oval cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'elongated oval cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                    rep_item(k).name = arg_list[j]))) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;

```


END_RULE;

5.2.5.163 free_form_interior_cutout_design_definition_has_max_items_elements_in_1

The free_form_interior_cutout_design_definition_has_max_items_elements_in_1 forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE free_form_interior_cutout_design_definition_has_max_items_elements_in_1
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
      SIZEOF (
        QUERY(
          temp2 <* USEDIN(temp1,
            'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
          | ((temp2\classification_assignment.role.name =
            'class membership') AND
            (temp_appl_class\classification_assignment.
            assigned_classification\group.name =
            'free form interior cutout design definition'))
          )
        ) > 0
    );

  WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
      SIZEOF(temp3.1) > items)) = 0;
END_RULE;
```

5.2.5.164 comp_rep_item_w_name_eq_bounding_curve

The comp_rep_item_w_name_eq_bounding_curve forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'bounding curve': VAL - value of the inherited name attribute of the compound_representation_item; 'free form interior cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_

representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_bounding_curve FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'free form interior cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'free form interior cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                         rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.165 interior_cutout_functional_definition_has_max_items_elements_in_1

The interior_cutout_functional_definition_has_max_items_elements_in_1 forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE interior_cutout_functional_definition_has_max_items_elements_in_1
FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |

```

```

        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
            | ((temp2\classification_assignment.role.name =
                'class membership') AND
                (temp_appl_class\classification_assignment.
                assigned_classification\group.name =
                'interior cutout functional definition'))
            )
        ) > 0
    );
WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.1) > items)) = 0;
END_RULE;

```

5.2.5.166 comp_rep_item_w_name_eq_the_function

The comp_rep_item_w_name_eq_the_function forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'the function': VAL - value of the inherited name attribute of the compound_representation_item; 'interior cutout functional definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_the_function FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'interior cutout functional definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
    temp_comp_rep_item.name =
    'interior cutout functional definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;

```

```

    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.167 inward_round_corner_design_definition_has_max_items_elements_in_3

The `inward_round_corner_design_definition_has_max_items_elements_in_3` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE inward_round_corner_design_definition_has_max_items_elements_in_3
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
      SIZEOF (
        QUERY(
          temp2 <* USEDIN(temp1,
            'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
          | ((temp2\classification_assignment.role.name =
            'class membership') AND
            (temp_appl_class\classification_assignment.
              assigned_classification\group.name =
                'inward round corner design definition'))
          )
        ) > 0
      );

  WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
      SIZEOF(temp3.3) > items)) = 0;
END_RULE;

```

5.2.5.168 comp_rep_item_w_name_eq_x_offset

The `comp_rep_item_w_name_eq_x_offset` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item`

has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x offset': VAL - value of the inherited name attribute of the compound_representation_item; 'inward round corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```
RULE comp_rep_item_w_name_eq_x_offset FOR (compound_representation_item);
```

```
LOCAL
```

```
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'inward round corner design definition';
    violation:  BOOLEAN := FALSE;
```

```
END_LOCAL;
```

```
creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'inward round corner design definition');
```

```
REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                        rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;
```

```
WHERE
```

```
    wr1: NOT violation;
```

```
END_RULE;
```

5.2.5.169 comp_rep_item_w_name_eq_y_offset

The comp_rep_item_w_name_eq_y_offset forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'y offset': VAL - value of the inherited name attribute of the compound_representation_item; 'inward round corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_y_offset FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'inward round corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name
                'inward round corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                         rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.170 comp_rep_item_w_name_eq_radius

The comp_rep_item_w_name_eq_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'radius': VAL - value of the inherited name attribute of the compound_representation_item 'inward round corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_radius FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'inward round corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |

```

```

        temp_comp_rep_item.name =
            'inward round corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
            rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.171 outward_round_corner_design_definition_has_max_items_elements_in_3

The `outward_round_corner_design_definition_has_max_items_elements_in_3` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE outward_round_corner_design_definition_has_max_items_elements_in_3
    FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
            | ((temp2\classification_assignment.role.name =
                'class membership') AND
                (temp_appl_class\classification_assignment.
                assigned_classification\group.name =
                'outward round corner design definition'))
            )
        ) > 0
    );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |

```

```

        SIZEOF(temp3.3) > items)) = 0;
END_RULE;

```

5.2.5.172 comp_rep_item_w_name_eq_x_offset

The `comp_rep_item_w_name_eq_x_offset` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'x offset': VAL - value of the inherited name attribute of the `compound_representation_item`; 'outward round corner design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_offset FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'outward round corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'outward round corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                        rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.173 comp_rep_item_w_name_eq_y_offset

The `comp_rep_item_w_name_eq_y_offset` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the

value given in the list. 'y offset': VAL - value of the inherited name attribute of the compound_representation_item; 'outward round corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_y_offset FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'outward round corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'outward round corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.174 comp_rep_item_w_name_eq_radius

The comp_rep_item_w_name_eq_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'radius': VAL - value of the inherited name attribute of the compound_representation_item; 'outward round corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_radius FOR (compound_representation_item);

```

```

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'outward round corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'outward round corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.175 part_edge_cutout_design_definition_has_max_items_elements_in_8

The part_edge_cutout_design_definition_has_max_items_elements_in_8 forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE part_edge_cutout_design_definition_has_max_items_elements_in_8
    FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
        QUERY(
            temp2 <* USEDIN(temp1,
            'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
        | ((temp2\classification_assignment.role.name =
        'class membership') AND
        (temp_appl_class\classification_assignment.
        assigned_classification\group.name =

```

```

        'part edge cutout design definition'))
      )
    ) > 0
  );
WHERE
  wr1: SIZEOF(QUERY(temp3 <* var_representation |
    SIZEOF(temp3.8) > items)) = 0;
END_RULE;

```

5.2.5.176 comp_rep_item_w_name_eq_R_left

The comp_rep_item_w_name_eq_R_left forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'R left': VAL - value of the inherited name attribute of the compound_representation_item; 'part edge cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_R_left FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'part edge cutout design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name =
    'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
      rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.177 comp_rep_item_w_name_eq_R_right

The `comp_rep_item_w_name_eq_R_right` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'R right': VAL - value of the inherited name attribute of the `compound_representation_item`, 'part edge cutout design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```
RULE comp_rep_item_w_name_eq_R_right FOR (compound_representation_item);
```

```
LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j]))) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;
```

5.2.5.178 comp_rep_item_w_name_eq_x_A

The `comp_rep_item_w_name_eq_x_A` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'x A': VAL - value of the inherited name attribute of the `compound_representation_item`; 'part edge cutout design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_A FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HIINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.179 comp_rep_item_w_name_eq_x_B

The comp_rep_item_w_name_eq_x_B forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x B': VAL - value of the inherited name attribute of the compound_representation_item; 'part edge cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_B FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |

```

```

        temp_comp_rep_item.name =
        'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                rep_item(k).name = arg_list[j]))) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.180 comp_rep_item_w_name_eq_x_C

The comp_rep_item_w_name_eq_x_C forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x C': VAL - value of the inherited name attribute of the compound_representation_item; 'part edge cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_C FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
    temp_comp_rep_item.name =
    'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                rep_item(k).name = arg_list[j]))) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.181 comp_rep_item_w_name_eq_x_D

The comp_rep_item_w_name_eq_x_D forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x D': VAL - value of the inherited name attribute of the compound_representation_item; 'part edge cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_D FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                        rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.182 comp_rep_item_w_name_eq_y_B

The comp_rep_item_w_name_eq_y_B forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value

given in the list. 'y B': VAL - value of the inherited name attribute of the compound_representation_item; 'part edge cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_y_B FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.183 comp_rep_item_w_name_eq_y_C

The comp_rep_item_w_name_eq_y_C forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'y C': VAL - value of the inherited name attribute of the compound_representation_item; 'part edge cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_y_C FOR (compound_representation_item);

LOCAL

```



```

    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'part edge cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'part edge cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.184 rectangular_cutback_corner_design_definition_has_max_items_-elements_in_4

A `rectangular_cutback_corner_design_definition_has_max_items_elements_in_4` forces the aggregate type attribute `The` that is an attribute in entity data type `T` that is of class `C` to have maximum `N` elements.

EXPRESS specification:

```

RULE rectangular_cutback_corner_design_definition_has_max_items_elements_in_4
  FOR (representation);

  LOCAL
    var_representation: BAG OF representation := [];
  END_LOCAL;

  var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
      QUERY(
        temp2 <* USEDIN(temp1,
        'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
        | ((temp2\classification_assignment.role.name =
        'class membership') AND
        (temp_appl_class\classification_assignment.
        assigned_classification\group.name =

```

```

        'rectangular cutback corner design definition'))
      )
    ) > 0
  );
WHERE
  wr1: SIZEOF(QUERY(temp3 <* var_representation |
    SIZEOF(temp3.4) > items)) = 0;
END_RULE;

```

5.2.5.185 comp_rep_item_w_name_eq_x_depth

The comp_rep_item_w_name_eq_x_depth forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x depth': VAL - value of the inherited name attribute of the compound_representation_item; 'rectangular cutback corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_depth FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'rectangular cutback corner design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name =
    'rectangular cutback corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
      rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.186 comp_rep_item_w_name_eq_y_depth

The comp_rep_item_w_name_eq_y_depth forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'y depth': VAL - value of the inherited name attribute of the compound_representation_item; 'rectangular cutback corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_y_depth FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'rectangular cutback corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'rectangular cutback corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                        rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.187 comp_rep_item_w_name_eq_radius

The comp_rep_item_w_name_eq_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'radius': VAL - value of the inherited name attribute of the compound_representation_item; 'rectangular cutback corner design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_radius FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'rectangular cutback corner design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'rectangular cutback corner design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HIINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.188 rectangular_cutout_design_definition_has_max_items_elements_in_2

The `rectangular_cutout_design_definition_has_max_items_elements_in_2` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE rectangular_cutout_design_definition_has_max_items_elements_in_2
FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
    SIZEOF (
        QUERY(
            temp2 <* USEDIN(temp1,

```

```

'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
| ((temp2\classification_assignment.role.name =
  'class membership') AND
  (temp_appl_class\classification_assignment.
  assigned_classification\group.name =
  'rectangular cutout design definition'))
  )
) > 0
);
WHERE
  wr1: SIZEOF(QUERY(temp3 <* var_representation |
    SIZEOF(temp3.2) > items)) = 0;
END_RULE;

```

5.2.5.189 comp_rep_item_w_name_eq_length_of

The comp_rep_item_w_name_eq_length_of forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'length of': VAL - value of the inherited name attribute of the compound_representation_item; 'rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_length_of FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'rectangular cutout design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name =
  'rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
      rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.190 comp_rep_item_w_name_eq_width

The comp_rep_item_w_name_eq_width forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'width': VAL - value of the inherited name attribute of the compound_representation_item; 'rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_width FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'rectangular cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.191 round_corner_rectangular_cutout_design_definition_has_max_items_elements_in_3

The round_corner_rectangular_cutout_design_definition_has_max_items_elements_in_3 forces the aggregate type attribute A that is an attribute in entity data type T that is of class

C to have maximum N elements.

EXPRESS specification:

```

RULE round_corner_rectangular_cutout_design_definition_has_max_items_
elements_in_3 FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
| ((temp2\classification_assignment.role.name =
'class membership') AND
(temp_appl_class\classification_assignment.
assigned_classification\group.name =
'round corner rectangular cutout design definition'))
            )
        ) > 0
    );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.3) > items)) = 0;
END_RULE;

```

5.2.5.192 comp_rep_item_w_name_eq_corner_radius

The comp_rep_item_w_name_eq_corner_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'corner radius': VAL - value of the inherited name attribute of the compound_representation_item; 'round corner rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_corner_radius FOR (compound_representation_item);

LOCAL

```

```

creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
arg_list:   LIST OF STRING :=
            'round corner rectangular cutout design definition';
violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'round corner rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j])) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.193 comp_rep_item_w_name_eq_length_of

The comp_rep_item_w_name_eq_length_of forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'length of': VAL - value of the inherited name attribute of the compound_representation_item; 'round corner rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_length_of FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING :=
            'round corner rectangular cutout design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'round corner rectangular cutout design definition');

```



```

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
      violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.194 comp_rep_item_w_name_eq_width

The comp_rep_item_w_name_eq_width forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'width': VAL - value of the inherited name attribute of the compound_representation_item; 'round corner rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_width FOR (compound_representation_item);

LOCAL
  creps:    BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list: LIST OF STRING :=
    'round corner rectangular cutout design definition';
  violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
               temp_comp_rep_item.name =
               'round corner rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
      violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.195 round_edge_rectangular_cutout_design_definition_has_max_items_elements_in_4

The round_edge_rectangular_cutout_design_definition_has_max_items_elements_in_4 forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE round_edge_rectangular_cutout_design_definition_has_max_items_elements_
in_4 FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
| ((temp2\classification_assignment.role.name =
'class membership') AND
(temp_appl_class\classification_assignment.
assigned_classification\group.name =
'round edge rectangular cutout design definition'))
            )
        ) > 0
    );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.4) > items)) = 0;
END_RULE;

```

5.2.5.196 comp_rep_item_w_name_eq_edge_radius

The comp_rep_item_w_name_eq_edge_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'edge radius': VAL - value of the inherited name attribute of the

compound_representation_item; 'round edge rectangular cutout design definition': arg_list
- list of names for the .name attributes of the representation_items pointed by compound_
representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_edge_radius FOR (compound_representation_item);

LOCAL
    creps:    BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:  LIST OF STRING :=
                'round edge rectangular cutout design definition';
    violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'round edge rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.197 comp_rep_item_w_name_eq_distance

The comp_rep_item_w_name_eq_distance forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'distance': VAL - value of the inherited name attribute of the compound_representation_item; 'round edge rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_distance FOR (compound_representation_item);

```

```

LOCAL
  creps:    BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list: LIST OF STRING :=
    'round edge rectangular cutout design definition';
  violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
  temp_comp_rep_item.name =
    'round edge rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
      violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
        rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.198 comp_rep_item_w_name_eq_length_of

The comp_rep_item_w_name_eq_length_of forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'length of': VAL - value of the inherited name attribute of the compound_representation_item; 'round edge rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_length_of FOR (compound_representation_item);

LOCAL
  creps:    BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list: LIST OF STRING :=
    'round edge rectangular cutout design definition';
  violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |

```

```

        temp_comp_rep_item.name =
            'round edge rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
            rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.199 comp_rep_item_w_name_eq_width

The comp_rep_item_w_name_eq_width forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'width': VAL - value of the inherited name attribute of the compound_representation_item; 'round edge rectangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_width FOR (compound_representation_item);

LOCAL
    creps:    BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list: LIST OF STRING :=
        'round edge rectangular cutout design definition';
    violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
    temp_comp_rep_item.name =
        'round edge rectangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
            rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;

```

```

    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.200 shear_bevel_design_definition_has_max_items_elements_in_4

The shear_bevel_design_definition_has_max_items_elements_in_4 forces the aggregate type attribute The that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE shear_bevel_design_definition_has_max_items_elements_in_4
    FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                    'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
                | ((temp2\classification_assignment.role.name =
                    'class membership') AND
                    (temp_appl_class\classification_assignment.
                    assigned_classification\group.name = 'shear bevel design definition'))
            )
        ) > 0
    );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.4) > items)) = 0;
END_RULE;

```

5.2.5.201 comp_rep_item_w_name_eq_x_z_angle

The comp_rep_item_w_name_eq_x_z_angle forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x z angle': VAL - value of the inherited name attribute of the com-

pound_representation_item; 'shear bevel design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_z_angle FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'shear bevel design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'shear bevel design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.202 comp_rep_item_w_name_eq_offset

The comp_rep_item_w_name_eq_offset forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'offset': VAL - value of the inherited name attribute of the compound_representation_item; 'shear bevel design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_offset FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'shear bevel design definition';

```

```

    violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'shear bevel design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
      violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.203 comp_rep_item_w_name_eq_x_y_angle

The comp_rep_item_w_name_eq_x_y_angle forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'x y angle': VAL - value of the inherited name attribute of the compound_representation_item; 'shear bevel design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_x_y_angle FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list:   LIST OF STRING := 'shear bevel design definition';
  violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name = 'shear bevel design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
      violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                   rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

```



```

END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.204 triangular_cutout_design_definition_has_max_items_elements_in_6

The `triangular_cutout_design_definition_has_max_items_elements_in_6` forces the aggregate type attribute A that is an attribute in entity data type T that is of class C to have maximum N elements.

EXPRESS specification:

```

RULE triangular_cutout_design_definition_has_max_items_elements_in_6
    FOR (representation);

LOCAL
    var_representation: BAG OF representation := [];
END_LOCAL;

var_representation := QUERY(
    temp1 <* representation |
        SIZEOF (
            QUERY(
                temp2 <* USEDIN(temp1,
                    'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
                | ((temp2\classification_assignment.role.name =
                    'class membership') AND
                    (temp_appl_class\classification_assignment.
                    assigned_classification\group.name =
                    'triangular cutout design definition'))
            )
        ) > 0
    );

WHERE
    wr1: SIZEOF(QUERY(temp3 <* var_representation |
        SIZEOF(temp3.6) > items)) = 0;
END_RULE;

```

5.2.5.205 comp_rep_item_w_name_eq_c2_x

The `comp_rep_item_w_name_eq_c2_x` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'c2 x': VAL - value of the inherited name attribute of the `compound_`

representation_item; 'triangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_c2_x FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'triangular cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'triangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.206 comp_rep_item_w_name_eq_c3_x

The comp_rep_item_w_name_eq_c3_x forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'c3 x': VAL - value of the inherited name attribute of the compound_representation_item; 'triangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_c3_x FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];

```

```

    arg_list: LIST OF STRING := 'triangular cutout design definition';
    violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'triangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                rep_item(k).name = arg_list[j]))) = 1);
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.207 comp_rep_item_w_name_eq_c3_y

The comp_rep_item_w_name_eq_c3_y forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'c3 y': VAL - value of the inherited name attribute of the compound_representation_item; 'triangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_c3_y FOR (compound_representation_item);

LOCAL
  creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
  arg_list: LIST OF STRING := 'triangular cutout design definition';
  violation: BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'triangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
  REPEAT k:=1 TO HINDEX(creps[i].item_element);
    violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |

```

```

                                rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.208 comp_rep_item_w_name_eq_c1_radius

The comp_rep_item_w_name_eq_c1_radius forces the item_element attribute of the compound_representation_item of which inherited 'name' attribute from representation_item has the value of VAL, to have a representation_item of which 'name' attribute has the value given in the list. 'c1 radius': VAL - value of the inherited name attribute of the compound_representation_item; 'triangular cutout design definition': arg_list - list of names for the .name attributes of the representation_items pointed by compound_representation_item.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_c1_radius FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'triangular cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'triangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
        REPEAT k:=1 TO HINDEX(creps[i].item_element);
            violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                           rep_item(k).name = arg_list[j])) = 1);
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.209 comp_rep_item_w_name_eq_c2_radius

The `comp_rep_item_w_name_eq_c2_radius` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'c2 radius': VAL - value of the inherited name attribute of the `compound_representation_item`; 'triangular cutout design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_c2_radius FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'triangular cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'triangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                        rep_item(k).name = arg_list[j])) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.210 comp_rep_item_w_name_eq_c3_radius

The `comp_rep_item_w_name_eq_c3_radius` forces the `item_element` attribute of the `compound_representation_item` of which inherited 'name' attribute from `representation_item` has the value of VAL, to have a `representation_item` of which 'name' attribute has the value given in the list. 'c3 radius': VAL - value of the inherited name attribute of the `compound_representation_item`; 'triangular cutout design definition': `arg_list` - list of names for the `.name` attributes of the `representation_items` pointed by `compound_representation_item`.

EXPRESS specification:

```

RULE comp_rep_item_w_name_eq_c3_radius FOR (compound_representation_item);

LOCAL
    creps:      BAG OF COMPOUND_REPRESENTATION_ITEM := [];
    arg_list:   LIST OF STRING := 'triangular cutout design definition';
    violation:  BOOLEAN := FALSE;
END_LOCAL;

creps := QUERY( temp_comp_rep_item <* compound_representation_item |
                temp_comp_rep_item.name =
                'triangular cutout design definition');

REPEAT i:=1 TO HIINDEX(creps) WHILE(NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE(NOT violation);
    REPEAT k:=1 TO HIINDEX(creps[i].item_element);
        violation := NOT (SIZEOF(QUERY(rep_item <* creps[i].item_element |
                                      rep_item(k).name = arg_list[j]))) = 1);
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.211 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

LOCAL
    version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects: BAG OF GENERIC := [];
    duplicate:        BOOLEAN := FALSE;
END_LOCAL;

version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
                    i.ROLE.NAME = 'version identifier');

```

```

REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
        duplicate := versionable_objects[i] :=: versionable_objects[j];
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT duplicate;
END_RULE;

```

5.2.5.212 ID_of_EXTENDED_PRODUCT_DEFINITION_SHAPE_with_- class_definition_pattern

The ID_of_EXTENDED_PRODUCT_DEFINITION_SHAPE_with_class_definition_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. EXTENDED_PRODUCT_DEFINITION_SHAPE: T - type having attribute A; 'definition': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_EXTENDED_PRODUCT_DEFINITION_SHAPE_with_class_definition_pattern
    FOR (CLASSIFICATION_ASSIGNMENT);

    LOCAL
        clfied_inst : SET OF GENERIC := [];
    END_LOCAL;

    REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
        IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name = 'definition')
            THEN
                clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
            END_IF;
    END_REPEAT;

    WHERE
        WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE *:))) = 0;
    END_RULE;

```

5.2.5.213 ID_of_EXTENDED_SHAPE_ASPECT_RELATIONSHIP_with_- class_definition_pattern

The ID_of_EXTENDED_SHAPE_ASPECT_RELATIONSHIP_with_class_definition_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. EXTENDED_SHAPE_ASPECT_RELATIONSHIP: T - type having attribute A; 'definition': C - class of T; ID: A - attribute of type T; *: *: P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_EXTENDED_SHAPE_ASPECT_RELATIONSHIP_with_class_definition_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name = 'definition')
    THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i < * clfied_inst | NOT(i.ID LIKE *:))) = 0;
END_RULE;

```

5.2.5.214 one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_- manufacturing_definition_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_definition_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'assembly manufacturing definition parameters': VAL - value of the .name attribute of the property_definition_representation; 'assembly method', 'assembly stage', 'mass', 'center of gravity': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_
  definition_parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['assembly method', 'assembly stage', 'mass',
                                'center of gravity'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'assembly manufacturing definition parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

**5.2.5.215 EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_-
assembly_manufacturing_definition_referenced_by_exactly_1_-
PROPERTY_DEFINITION_REPRESENTATION_via_-
DEFINITION_w_NAME_eq_assembly_manufacturing_definition_-
parameters**

The EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_assembly_manufacturing_definition_parameters gets all classification_assignment instances with id 'assembly manufacturing definition' for all instances of EXTENDED_PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a EXTENDED_PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "as-

sembly manufacturing definition parameters” check if their number equals 1.

EXPRESS specification:

```

RULE EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_
  definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_
  via_DEFINITION_w_NAME_eq_assembly_manufacturing_definition_parameters
  FOR(EXTENDED_PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
    APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF EXTENDED_PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'assembly manufacturing definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'assembly manufacturing definition parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.216 one_item_in_rep_if_used_in_pdr_w_name_eq_assembly- manufacturing_position_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_position_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'assembly manufacturing position parameters': VAL - value of the .name attribute of the property_definition_representation; 'assembly

footprint','orientation': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_position_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['assembly footprint','orientation'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'assembly manufacturing position parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.217 SHAPE_ASPECT_w_class_id_assembly_manufacturing_position_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_assembly_-manufacturing_position_parameters

The SHAPE_ASPECT_w_class_id_assembly_manufacturing_position_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_assembly_manufacturing_position_parameters gets all classification_assignment instances with id 'assembly manufacturing position' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINI-

TION_REPRESENTATION instances whose attribute NAME has the value "assembly manufacturing position parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_assembly_manufacturing_position_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  assembly_manufacturing_position_parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'assembly manufacturing position');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
  'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'assembly manufacturing position parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.218 EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_ **plate_manufacturing_definition_referenced_by_exactly_1_** **PROPERTY_DEFINITION_REPRESENTATION_via_** **DEFINITION_w_NAME_eq_outer_contour_representation**

The EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_outer_contour_representation gets all classification_assignment in-

stances with id 'plate manufacturing definition' for all instances of EXTENDED_PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a EXTENDED_PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "outer contour representation" check if their number equals 1.

EXPRESS specification:

```

RULE EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_
  definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_
  via_DEFINITION_w_NAME_eq_outer_contour_representation
  FOR(EXTENDED_PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF EXTENDED_PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'plate manufacturing definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'outer contour representation')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.219 EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_outer_contour_via_OF_SHAPE

The EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_outer_contour_via_OF_SHAPE for all instances of EXTENDED_PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a EXTENDED_PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'outer contour' check if their number equals 1.

EXPRESS specification:

```

RULE EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_
  definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_outer_contour_
  via_OF_SHAPE
  FOR(EXTENDED_PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF EXTENDED_PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'plate manufacturing definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'SHAPE_ASPECT' + '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'outer contour' IN WHICH_CLASS(i))) < 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.220 one_item_in_rep_if_used_in_pdr_w_name_eq_structural_added_material_feature_design_definition_parameter_representation

The one_item_in_rep_if_used_in_pdr_w_name_eq_structural_added_material_feature_design_definition_parameter_representation forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'structural added material feature design definition parameter representation': VAL - value of the .name attribute of the property_definition_representation; 'added_material_length': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_structural_added_material_feature_design_
definition_parameter_representation FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['added_material_length'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'structural added material
  feature design definition parameter representation')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.221 optional_occurrence_of_attributes_for_structural_weld_shrinkage-allowance_feature_design_definition_parameter_representation

The optional_occurrence_of_attributes_for_structural_weld_shrinkage_allowance_feature_design_definition_parameter_representation forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'structural weld shrinkage allowance feature design definition parameter representation': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'longitudinal allowance', 'transverse allowance', 'vertical allowance': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```
RULE optional_occurrence_of_attributes_for_structural_weld_shrinkage_allowance_
feature_design_definition_parameter_representation FOR (representation);
```

```
LOCAL
```

```
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['longitudinal allowance',
                              'transverse allowance', 'vertical allowance'];
```

```
  found: BOOLEAN := FALSE;
```

```
END_LOCAL;
```

```
reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'structural weld shrinkage allowance
  feature design definition parameter representation')
      )
    ) > 0
);
```

```
REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;
```

```
WHERE
```

```
  wr1: NOT found;
```

```
END_RULE;
```


5.2.5.222 EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_- weld_manufacturing_definition_referenced_by_exactly_1_SHAPE_- ASPECT_w_class_id_torch_vector_via_OF_SHAPE

The EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_weld_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_torch_vector_via_OF_SHAPE gets all classification_assignment instances with id 'weld manufacturing definition' for all instances of EXTENDED_PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a EXTENDED_PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'torch vector' check if their number equals 1.

EXPRESS specification:

```

RULE EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_weld_manufacturing_definition_
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_torch_vector_via_OF_SHAPE
FOR(EXTENDED_PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_
ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF EXTENDED_PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'weld manufacturing definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'SHAPE_ASPECT' + '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'torch vector' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.223 one_item_in_rep_if_used_in_pdr_w_name_eq_weld_manufacturing_definition_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_weld_manufacturing_definition_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'weld manufacturing definition parameters': VAL - value of the .name attribute of the property_definition_representation; 'position', 'process', 'degree of automations', 'number of weld passes', 'welding deposition sequences', 'welding environment', 'tack weld used': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_weld_manufacturing_definition_parameters
  FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['position', 'process', 'degree of automations',
                                'number of weld passes',
                                'welding deposition sequences',
                                'welding environment', 'tack weld used'];

  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'weld manufacturing definition parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.224 EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_- assembly_manufacturing_definition_referenced_by_exactly_1_- SHAPE_ASPECT_w_class_id_centre_of_gravity_via_OF_SHAPE

The EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_centre_of_gravity_via_OF_SHAPE gets all classification_assignment instances with id 'assembly manufacturing definition' for all instances of EXTENDED_PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a EXTENDED_PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'centre of gravity' check if their number equals 1.

EXPRESS specification:

```

RULE EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_
definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_centre_of_
gravity_via_OF_SHAPE
FOR(EXTENDED_PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF EXTENDED_PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME =
                'assembly manufacturing definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                    'SHAPE_ASPECT' + '.' + 'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                'centre of gravity' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.225 one_item_in_rep_if_used_in_pdr_w_name_eq_welding_sequence- representation

The `one_item_in_rep_if_used_in_pdr_w_name_eq_welding_sequence_representation` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` whose 'name' attribute has a value VAL. 'welding sequence representation': VAL - value of the .name attribute of the `property_definition_representation`; 'name', 'start offset', 'end offset', 'direction': arg_list - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_welding_sequence_representation
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['name', 'start offset', 'end offset',
                                'direction'];

    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'welding sequence representation')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.226 **dsc_rep_item_w_n_user_def_function_if_dsc_rep_item_w_n_function_and_d_user_defined**

The `dsc_rep_item_w_n_user_def_function_if_dsc_rep_item_w_n_function_and_d_user_defined` forces a representation that has a `descriptive_representation_item` with name N1 and description D1 to have another `descriptive_representation_item` with name N2. 'function': N1 - value of the `.name` attribute of the `descriptive_representation_item`; 'user defined': D1 - value of the `.description` attribute of the `descriptive_representation_item`; 'user def function': N2 - value of the `.name` attribute of the other `descriptive_representation_item`.

EXPRESS specification:

```

RULE dsc_rep_item_w_n_user_def_function_if_dsc_rep_item_w_n_function_and_d_
    user_defined FOR (REPRESENTATION);

LOCAL
    rep_set:    SET OF representation := [];
    violation:  BOOLEAN := FALSE;
END_LOCAL;

REPEAT i := 1 TO HIINDEX(REPRESENTATION) WHILE NOT violation;
    violation := (SIZEOF(QUERY(r <* REPRESENTATION[i].ITEMS |
        (r.NAME = 'function') AND (r.DESRIPTION = 'user defined')))) > 0) AND
        (SIZEOF(QUERY(r <* REPRESENTATION[i].ITEMS |
            (r.NAME = 'user def function')))) = 0);
END_REPEAT;

WHERE
    WR1: NOT violation;
END_RULE;

```

5.2.5.227 **ID_of_PRODUCT_DEFINITION_with_class_definable_object_pattern**

The `ID_of_PRODUCT_DEFINITION_with_class_definable_object_pattern` forces string type attribute A of type T to match the pattern P if the instance of T is of class C. `PRODUCT_DEFINITION`: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; '*. *': P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_PRODUCT_DEFINITION_with_class_definable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDEX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.
      name = 'definable object') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE '*.*')) = 0;
  END_RULE;

```

5.2.5.228 same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_structural_part_symmetry_relationship

The same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_structural_part_symmetry_relationship forces the relating and the related side of a relationship of type T_relationship to be of the same class if T_relationship is of class C. PRODUCT_DEFINITION: T - type that is related by T_relationship; 'structural part symmetry relationship': C - class of T_relationship.

EXPRESS specification:

```

RULE same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_
  class_structural_part_symmetry_relationship
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    cl_ass:      SET OF CLASSIFICATION_ASSIGNMENT := [];
    clfied_inst: SET OF PRODUCT_DEFINITION_RELATIONSHIP := [];
  END_LOCAL;

  cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME =
      'structural part symmetry relationship');

  REPEAT i := 1 TO HIINDEX(cl_ass);
    clfied_inst := clfied_inst + cl_ass[i].items;
  END_REPEAT;

```

```

WHERE
  WR1: SIZEOF(WHICH_CLASS(i.RELATING_PRODUCT_DEFINITION) -
    WHICH_CLASS(i.RELATED_PRODUCT_DEFINITION)) = 0;
END_RULE;

```

5.2.5.229 ID_of_EXTENDED_PROPERTY_DEFINITION_with_class_definition_pattern

The ID_of_EXTENDED_PROPERTY_DEFINITION_with_class_definition_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. EXTENDED_PROPERTY_DEFINITION: T - type having attribute A; 'definition': C - class of T; ID: A - attribute of type T; '*. *': P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_EXTENDED_PROPERTY_DEFINITION_with_class_definition_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clfied_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name = 'definition') THEN
      clfied_inst := clfied_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clfied_inst | NOT(i.ID LIKE '*. *')) = 0;
END_RULE;

```

5.2.5.230 ID_of_PRODUCT_DEFINITION_RELATIONSHIP_with_class_definable_object_pattern

The ID_of_PRODUCT_DEFINITION_RELATIONSHIP_with_class_definable_object_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C. PRODUCT_DEFINITION_RELATIONSHIP: T - type having attribute A; 'definable object': C - class of T; ID: A - attribute of type T; '*. *': P - pattern that T.A shall match.

EXPRESS specification:

```

RULE ID_of_PRODUCT_DEFINITION_RELATIONSHIP_with_class_definable_object_pattern
  FOR (CLASSIFICATION_ASSIGNMENT);

  LOCAL
    clified_inst : SET OF GENERIC := [];
  END_LOCAL;

  REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.
      name = 'definable object') THEN
      clified_inst := clified_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
  END_REPEAT;

  WHERE
    WR1: SIZEOF(QUERY(i <* clified_inst | NOT(i.ID LIKE '*.*')) = 0;
  END_RULE;

```

5.2.5.231 one_item_in_rep_if_used_in_pdr_w_name_eq_angle_bar_cross-section_design_parameters

A `one_item_in_rep_if_used_in_pdr_w_name_eq_angle_bar_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` whose 'name' attribute has a value VAL. 'angle bar cross section design parameters': VAL - value of the .name attribute of the `property_definition_representation`; 'depth', 'width', 'radius', 'thk': arg_list - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that representation..

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_angle_bar_cross_section_design_
  parameters FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['depth', 'width', 'radius', 'thk'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (

```



```

        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'angle bar cross section design parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.232 one_item_in_rep_if_used_in_pdr_w_name_eq_bulb_flat_cross-section_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_bulb_flat_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'bulb flat cross section design parameters': VAL - value of the .name attribute of the property_definition_representation, 'depth', 'bulb width', 'bulb radius', 'tw': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_bulb_flat_cross_section_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['depth', 'bulb width', 'bulb radius', 'tw'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,

```

```

'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'bulb flat cross section design parameters')
    )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.233 one_item_in_rep_if_used_in_pdr_w_name_eq_channel_cross_section_- design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_channel_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'channel cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'depth', 'width', 'web thk', 'flange thk', 'radius', 'k': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_channel_cross_section_design_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['depth', 'width', 'web thk', 'flange thk',
                                'radius', 'k'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')

```

```

| (temp_prop_def_rep.name = 'channel cross section design parameters')
    )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.234 one_item_in_rep_if_used_in_pdr_w_name_eq_circular_hollow_- profile_cross_section_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_circular_hollow_profile_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'circular hollow profile cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'outer diameter', 'inner diameter': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_circular_hollow_profile_
  cross_section_design_parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['outer diameter', 'inner diameter'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
          'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'circular hollow profile cross section design parameters')
      )
    )
)

```

```

        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.235 one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_part- design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_part_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'corrugated part design parameters': VAL - value of the .name attribute of the property_definition_representation; 'mirrored', 'offset', 'repetition', 'thickness': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_part_design_parameters
    FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['mirrored', 'offset', 'repetition',
                                'thickness'];

    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'corrugated part design parameters')
            )
        ) > 0

```

```

    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.236 optional_occurrence_of_attributes_for_corrugated_part_design_parameters

The `optional_occurrence_of_attributes_for_corrugated_part_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'corrugated part design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'mass': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_corrugated_part_design_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['mass'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'corrugated part design parameters')
      )
    ) > 0
  );

```

```

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j]))) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.237 one_item_in_rep_if_used_in_pdr_w_name_eq_corrugation_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_corrugation_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` whose 'name' attribute has a value VAL. 'corrugation design parameters': VAL - value of the .name attribute of the `property_definition_representation`; 'depth', 'flat width 1', 'flat width 2', 'slope width': arg_list - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_corrugation_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['depth', 'flat width 1', 'flat width 2', 'slope width'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'corrugation design parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);

```

```

        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.238 optional_occurrence_of_attributes_for_corrugation_design-parameters

The optional_occurrence_of_attributes_for_corrugation_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'corrugation design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'radius 1', 'radius 2': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_corrugation_design_parameters
    FOR (representation);

    LOCAL
        reps: BAG OF REPRESENTATION := [];
        arg_list: LIST OF STRING := ['radius 1', 'radius 2'];
        found: BOOLEAN := FALSE;
    END_LOCAL;

    reps := QUERY(
        temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
                'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'corrugation design parameters')
            )
        ) > 0
    );

    REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name=arg_list[j])) > 1);

```

```

    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.239 one_item_in_rep_if_used_in_pdr_w_name_eq_explicit_profile_cross-section_shape

The `one_item_in_rep_if_used_in_pdr_w_name_eq_explicit_profile_cross-section_shape` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` whose 'name' attribute has a value VAL. 'explicit profile cross section shape': VAL - value of the .name attribute of the `property_definition_representation`; 'cross section geometry', 'local coordinate system': arg_list - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_explicit_profile_cross_section_
    shape FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['cross section geometry',
                                'local coordinate system'];

    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name =
                    'explicit profile cross section shape')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);

```



```

    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.240 one_item_in_rep_if_used_in_pdr_w_name_eq_flanged_plate_cross-section_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_flanged_plate_cross-section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'flanged plate cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'depth', 'width', 'radius', 'thk': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_flanged_plate_cross_section_
    design_parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['depth', 'width', 'radius', 'thk'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'flanged plate cross section design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.241 one_item_in_rep_if_used_in_pdr_w_name_eq_flat_bar_cross_section_- design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_flat_bar_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list (...) exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'flat bar cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'depth', 'width': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_flat_bar_cross_section_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['depth', 'width'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'flat bar cross section design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.242 one_item_in_rep_if_used_in_pdr_w_name_eq_plate_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_plate_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'plate design parameters': VAL - value of the .name attribute of the property_definition_representation; 'material offset', 'thickness': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_plate_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['material offset', 'thickness'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
            'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'plate design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.243 optional_occurrence_of_attributes_for_plate_design_parameters

The `optional_occurrence_of_attributes_for_plate_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'plate design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'mass': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_plate_design_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['mass'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
            'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'plate design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
  found := (SIZEOF(QUERY(rep_item <* reps[i].items |
    rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.244 optional_occurrence_of_attributes_for_profile_boundary_relationship_design_parameters

The optional_occurrence_of_attributes_for_profile_boundary_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'profile boundary relationship design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_profile_boundary_relationship_
  design_parameters FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['displacement'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'profile boundary relationship design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.245 one_item_in_rep_if_used_in_pdr_w_name_eq_profile_curve_trace_line_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_profile_curve_trace_line_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'profile curve trace line design parameters': VAL - value of the .name attribute of the property_definition_representation; 'curve': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_profile_curve_trace_line_design_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['curve'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'profile curve trace line design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.246 optional_occurrence_of_attributes_for_profile_curve_trace_line_- design_parameters

The optional_occurrence_of_attributes_for_profile_curve_trace_line_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'profile curve trace line design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_profile_curve_trace_line_design_
parameters FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['displacement'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'profile curve trace line design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.247 one_item_in_rep_if_used_in_pdr_w_name_eq_profile_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_profile_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'profile design parameters': VAL - value of the .name attribute of the property_definition_representation; 'mirrored': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_profile_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['mirrored'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'profile design parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```


5.2.5.248 optional_occurrence_of_attributes_for_profile_design_parameters

The `optional_occurrence_of_attributes_for_profile_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list (...) zero or one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL.

'profile design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'mass': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_profile_design_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['mass'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'profile design parameters')
        )
      ) > 0
  );

  iterate over all representations found above. Stop, if for one of
  them the names of its representation_items are duplicated.

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
  found := (SIZEOF(QUERY(rep_item <* reps[i].items |
    rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.249 optional_occurrence_of_attributes_for_profile_trace_line_- relationship_design_parameters

The optional_occurrence_of_attributes_for_profile_trace_line_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'profile trace line relationship design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement', 'auxiliary line': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_profile_trace_line_relationship_
  design_parameters FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['displacement', 'auxiliary line'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'profile trace line relationship design
  parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j]))) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;
```

5.2.5.250 one_item_in_rep_if_used_in_pdr_w_name_eq_round_bar_cross_section_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_round_bar_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'round bar cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'diameter': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_round_bar_cross_section_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['diameter'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'round bar cross section design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.251 one_item_in_rep_if_used_in_pdr_w_name_eq_section_properties

The `one_item_in_rep_if_used_in_pdr_w_name_eq_section_properties` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list if the representation is the 'used-representation' in a `property_definition_representation` whose 'name' attribute has a value VAL. 'section properties': VAL - value of the .name attribute of the `property_definition_representation`; 'nominal mass per length', 'area', 'na u', 'na v', 'moi u', 'moi v', 'moi uv', 'tr': arg_list - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_section_properties
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['nominal mass per length', 'area', 'na u',
                                'na v', 'moi u', 'moi v', 'moi uv', 'tr'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'section properties')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.252 one_item_in_rep_if_used_in_pdr_w_name_eq_square_tube_cross_section_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_square_tube_cross_section_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'square tube cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'depth', 'width', 'wall thk': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_square_tube_cross_section_design_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['depth', 'width', 'wall thk'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'square tube cross section design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.253 one_item_in_rep_if_used_in_pdr_w_name_eq_twist_location_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_twist_location_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'twist location design parameters': VAL - value of the .name attribute of the property_definition_representation; 'location', 'direction': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_twist_location_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['location', 'direction'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'twist location design parameters')
        )
      ) > 0
    );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.254 one_item_in_rep_if_used_in_pdr_w_name_eq_t_bar_cross_section_- design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_t_bar_cross_section_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 't bar cross section design parameters': VAL - value of the .name attribute of the property_definition_representation; 'depth', 'width', 'web thk', 'flange thk', 'radius', 'k': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_t_bar_cross_section_design_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['depth', 'width', 'web thk', 'flange thk',
                                'radius', 'k'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 't bar cross section design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.255 one_item_in_rep_if_used_in_pdr_w_name_eq_w_shape_cross_section

The `one_item_in_rep_if_used_in_pdr_w_name_eq_w_shape_cross_section` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'w shape cross section': VAL - value of the .name attribute of the property_definition_representation; 'depth', 'width', 'web thk', 'flange thk', 'radius', 'k': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_w_shape_cross_section
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['depth', 'width', 'web thk', 'flange thk',
                                'radius', 'k'];

    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'w shape cross section')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```


5.2.5.256 PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_- design_definition_referenced_by_exactly_2_SHAPE_ASPECT_w_- class_id_border_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_exactly_2_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE gets all classification_assignment instances with id 'corrugated part design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'border' check if their number equals 2.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_
referenced_by_exactly_2_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME =
                 'corrugated part design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' + '.' +
                  'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                'border' IN WHICH_CLASS(i))) = 2);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.257 PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_- design_definition_referenced_by_exactly_1_PROPERTY_- DEFINITION_REPRESENTATION_via_DEFINITION_w_- NAME_eq_corrugated_part_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_part_design_parameters gets all classification_assignment instances with id 'corrugated part design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "corrugated part design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_corrugated_part_design_parameters
FOR (PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'corrugated part design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'corrugated part design parameters')) = 1);
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.258 PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_- design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_- class_id_corrugation_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_corrugation_via_OF_SHAPE gets all classification_assignment instances with id 'corrugated part design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'corrugation' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_corrugation_via_OF_SHAPE
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME =
                'corrugated part design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
                    '.' + 'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                'corrugation' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

WHERE
    wr1: NOT violation;

```

END_RULE;

5.2.5.259 SHAPE_ASPECT_w_class_id_corrugation_referenced_exactly_- once_by_PROPERTY_DEFINITION_REPRESENTATION

The SHAPE_ASPECT_w_class_id_corrugation_referenced_exactly_once_by_PROPERTY_DEFINITION_REPRESENTATION gets all APPLIED_CLASSIFICATION_ASSIGNMENT instances with id 'corrugation'.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_corrugation_referenced_exactly_once_by_PROPERTY_
  DEFINITION_REPRESENTATION
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF APPLIED_CLASSIFICATION_ASSIGNMENT := [];
  bag_1: BAG OF SHAPE_ASPECT := [];
  bag_2: BAG OF SHAPE_ASPECT := [];
  multiple_references: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'corrugation');

REPEAT i := 1 TO HIINDEX(SHAPE_ASPECT);
  bag_1 := bag_1 + USEDIN(SHAPE_ASPECT[i], '');
END_REPEAT;

REPEAT i := 1 TO HIINDEX(bag_1);
  REPEAT j := 1 TO HIINDEX(c_a_set);
    IF (bag_1[i] IN c_a_set[j].items) THEN
      bag_2 := bag_2 + bag_1[i];
    END_IF;
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(bag_2) WHILE NOT multiple_references;
  REPEAT j := i + 1 TO HIINDEX(bag_2) WHILE NOT multiple_references;
    IF (bag_2[i] :=: bag_2[j]) THEN
      multiple_references := TRUE;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT multiple_references;

```

END_RULE;

5.2.5.260 SHAPE_ASPECT_w_class_id_angle_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_angle_bar_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_angle_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_angle_bar_cross_section_design_parameters gets all classification_assignment instances with id 'angle bar cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "angle bar cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_angle_bar_cross_section_referenced_by_exactly_1_
  PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_angle_bar_cross_
  section_design_parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'angle bar cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'angle bar cross section design parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.261 SHAPE_ASPECT_w_class_id_bulbflat_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_bulbflat_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_bulbflat_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_bulbflat_cross_section_design_parameters gets all classification_assignment instances with id 'bulbflat cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "bulbflat cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_bulbflat_cross_section_referenced_by_exactly_
1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_bulbflat_
cross_section_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF SHAPE_ASPECT := [];
    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'bulbflat cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
        'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
        i.NAME = 'bulbflat cross section design parameters')) = 1);

```

```

END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.262 SHAPE_ASPECT_w_class_id_channel_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_channel_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_channel_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_channel_cross_section_design_parameters gets all classification_assignment instances with id 'channel cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "channel cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_channel_cross_section_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_channel_cross_
section_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF SHAPE_ASPECT := [];
    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'channel cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
        'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');

```

```

violation := NOT (SIZEOF(QUERY(i <* t2_set |
                               i.NAME = 'channel cross section design parameters')) = 1);
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.263 SHAPE_ASPECT_w_class_id_circular_hollow_cross_section_- referenced_by_exactly_1_PROPERTY_DEFINITION_- REPRESENTATION_via_DEFINITION_w_NAME_eq_circular_- hollow_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_circular_hollow_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_circular_hollow_cross_section_design_parameters gets all classification_assignment instances with id 'circular hollow cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "circular hollow cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_circular_hollow_cross_section_referenced_by_
    exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
    circular_hollow_cross_section_design_parameters
    FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
        APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set:  SET OF SHAPE_ASPECT := [];
    t2_set:  SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

WHERE
    wr1: NOT violation;
END_RULE;

```


5.2.5.264 SHAPE_ASPECT_w_class_id_explicit_profile_cross_section_-referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_explicit_profile_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_explicit_profile_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_explicit_profile_cross_section_design_parameters gets all classification_assignment instances with id 'explicit profile cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value " explicit profile cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_explicit_profile_cross_section_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  explicit_profile_cross_section_design_parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'explicit profile cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = ' explicit profile cross section design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.265 SHAPE_ASPECT_w_class_id_flat_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flat_bar_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_flat_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flat_bar_cross_section_design_parameters gets all classification_assignment instances with id 'flat bar cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "flat bar cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_flat_bar_cross_section_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flat_bar_cross_
section_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'flat bar cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'flat bar cross section design parameters')) = 1);
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.266 SHAPE_ASPECT_w_class_id_flanged_plate_cross_section_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_flanged_plate_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_flanged_plate_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flanged_plate_cross_section_design_parameters gets all classification_assignment instances with id 'flanged plate cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "flanged plate cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_flanged_plate_cross_section_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_flanged_plate_cross_
section_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_CLASSIFICATION_
ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF SHAPE_ASPECT := [];
    t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'flanged plate cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                i.NAME = 'flanged plate cross section design parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.267 PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE gets all classification_assignment instances with id 'plate design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'border' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_
    by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE
    FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'plate design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
        '.' + 'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
        'border' IN WHICH_CLASS(i))) < 1);
END_REPEAT;

WHERE
    wr1: NOT violation;

```

END_RULE;

5.2.5.268 **PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_design_parameters**

The PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_design_parameters gets all classification_assignment instances with id 'plate design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "plate design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_
  by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  plate_design_parameters
  FOR(PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
    APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'plate design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'plate design parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.269 PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE gets all classification_assignment instances with id 'plate design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'moulded surface' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_
    by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE
    FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'plate design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
        '.' + 'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
        'moulded surface' IN WHICH_CLASS(i))) > 1);
END_REPEAT;

WHERE
    wr1: NOT violation;

```

END_RULE;

5.2.5.270 one_item_in_rep_if_used_in_pdr_w_name_eq_plate_function_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_plate_function_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'plate function parameters': VAL - value of the .name attribute of the property_definition_representation; 'function': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_plate_function_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['function'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
            'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'plate function parameters')
        )
      ) > 0
    );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.271 PROPERTY_DEFINITION_w_class_id_plate_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_function_parameters

The PROPERTY_DEFINITION_w_class_id_plate_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_function_parameters gets all classification_assignment instances with id 'plate functional definition' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "plate function parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_plate_functional_definition_referenced_
  by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  plate_function_parameters
  FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'plate functional definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'plate function parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;

```


END_RULE;

5.2.5.272 SHAPE_ASPECT_w_class_id_profile_cross_section_referenced_by_at_most_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_section_properties

The SHAPE_ASPECT_w_class_id_profile_cross_section_referenced_by_at_most_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_section_properties gets all classification_assignment instances with id 'profile cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value 'section properties' check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_profile_cross_section_referenced_by_at_most_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_section_properties
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'profile cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'section properties')) > 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.273 SHAPE_ASPECT_w_class_id_profile_curve_trace_line_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_curve_trace_line_design_parameters

The SHAPE_ASPECT_w_class_id_profile_curve_trace_line_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_curve_trace_line_design_parameters gets all classification_assignment instances with id 'profile curve trace line' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "profile curve trace line design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_profile_curve_trace_line_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_curve_
trace_line_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'profile curve trace line');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'profile curve trace line design parameters')) = 1);
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.274 PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_N_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_N_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE gets all classification_assignment instances with id 'profile design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'border' check if their number equals N.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_
  by_exactly_N_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'profile design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
    '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'border' IN WHICH_CLASS(i))) = 0) AND
    NOT (SIZEOF(QUERY(i <* t2_set |
    'border' IN WHICH_CLASS(i))) = 2);
END_REPEAT;

WHERE
  wr1: NOT violation;

```

END_RULE;

5.2.5.275 **PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_design_parameters**

The PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_design_parameters gets all classification_assignment instances with id 'profile design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "profile design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_
  by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  profile_design_parameters
  FOR(PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
    APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'profile design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'profile design parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.276 PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_profile_curve_trace_line_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_profile_curve_trace_line_via_OF_SHAPE gets all classification_assignment instances with id 'profile design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'profile curve trace line' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_
    by_exactly_1_SHAPE_ASPECT_w_class_id_profile_curve_trace_line_via_OF_SHAPE
    FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF SHAPE_ASPECT := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'profile design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
        '.' + 'OF_SHAPE');
    violation := NOT (SIZEOF(QUERY(i <* t2_set |
        'profile curve trace line' IN WHICH_CLASS(i))) > 1);
END_REPEAT;

WHERE
    wr1: NOT violation;

```

END_RULE;

5.2.5.277 **PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_profile_cross_section_via_OF_SHAPE**

The PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_profile_cross_section_via_OF_SHAPE gets all classification_assignment instances with id 'profile design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'profile cross section' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_
  by_exactly_1_SHAPE_ASPECT_w_class_id_profile_cross_section_via_OF_SHAPE
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'profile design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
    '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'profile cross section' IN WHICH_CLASS(i))) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.278 one_item_in_rep_if_used_in_pdr_w_name_eq_profile_function_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_profile_function_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'profile function parameters': VAL - value of the .name attribute of the property_definition_representation; 'function': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_profile_function_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['function'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'profile function parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.279 PROPERTY_DEFINITION_w_class_id_profile_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_function_parameters

The PROPERTY_DEFINITION_w_class_id_profile_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_profile_function_parameters gets all classification_assignment instances with id 'profile functional definition' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "profile function parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_profile_functional_definition_referenced_
  by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  profile_function_parameters
  FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'profile functional definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'profile function parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;

```


END_RULE;

5.2.5.280 SHAPE_ASPECT_w_class_id_round_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_round_bar_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_round_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_round_bar_cross_section_design_parameters gets all classification_assignment instances with id 'round bar cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "round bar cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_round_bar_cross_section_referenced_by_exactly_1_
  PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_round_bar_
  cross_section_design_parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'round bar cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'round bar cross section design parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.281 optional_occurrence_of_attributes_for_plate_function_parameters

The `optional_occurrence_of_attributes_for_plate_function_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'plate function parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'user def function': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_plate_function_parameters
    FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['user def function'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'plate function parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.282 optional_occurrence_of_attributes_for_profile_function_parameters

The `optional_occurrence_of_attributes_for_profile_function_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one `representation_item` who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` who's 'name' attribute has a value VAL. 'profile function parameters': VAL - value of the .name attribute of the `property_definition_representation` that uses the representation; 'user def function': arg_list - list of names for the .name attributes of the `representation_items` that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_profile_function_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['user def function'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'profile function parameters')
        )
      ) > 0
    );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
      found := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.283 one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_- symmetry_relationship_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_symmetry_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'structural part symmetry relationship design parameters': VAL - value of the .name attribute of the property_definition_representation; 'mirroring plane': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_symmetry_
relationship_design_parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['mirroring plane'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'structural part symmetry relationship
  design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j]))) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;
```

5.2.5.284 SHAPE_ASPECT_w_class_id_twist_location_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_twist_location_design_parameters

The SHAPE_ASPECT_w_class_id_twist_location_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_twist_location_design_parameters gets all classification_assignment instances with id 'twist location' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "twist location design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_twist_location_referenced_by_exactly_1_PROPERTY_
  DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_twist_location_design_
  parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'twist location');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'twist location design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.285 SHAPE_ASPECT_w_class_id_t_bar_cross_section_referenced_by_- exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_- DEFINITION_w_NAME_eq_t_bar_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_t_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_t_bar_cross_section_design_parameters gets all classification_assignment instances with id 't bar cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "t bar cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_t_bar_cross_section_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_t_bar_cross_
section_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 't bar cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 't bar cross section design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.286 SHAPE_ASPECT_w_class_id_w_shape_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_w_shape_cross_section_design_parameters

The SHAPE_ASPECT_w_class_id_w_shape_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_w_shape_cross_section_design_parameters gets all classification_assignment instances with id 'w shape cross section' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "w shape cross section design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_w_shape_cross_section_referenced_by_exactly_1_
PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_w_shape_
cross_section_design_parameters
FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'w shape cross section');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
                  'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
                                i.NAME = 'w shape cross section design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;

```

END_RULE;

5.2.5.287 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
  FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

  LOCAL
    version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects: BAG OF GENERIC := [];
    duplicate:        BOOLEAN := FALSE;
  END_LOCAL;

  version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
    i.ROLE.NAME = 'version identifier');

  REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
  END_REPEAT;

  REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
      duplicate := versionable_objects[i] :=: versionable_objects[j];
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT duplicate;
END_RULE;

```

5.2.5.288 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object
  FOR (APPLIED_IDENTIFICATION_ASSIGNMENT);

  LOCAL
    version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
    versionable_objects: BAG OF GENERIC := [];
    duplicate:        BOOLEAN := FALSE;
  END_LOCAL;

  version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
    i.ROLE.NAME = 'version identifier');

  REPEAT i := 1 TO HIINDEX(version_ids);
    versionable_objects := versionable_objects + version_ids[i].items;
  END_REPEAT;

  REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
      duplicate := versionable_objects[i] :=: versionable_objects[j];
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT duplicate;
END_RULE;

```

5.2.5.289 dsc_rep_item_w_n_user_defined_tightness_if_dsc_rep_item_w_n_tightness_and_d_user_defined

The dsc_rep_item_w_n_user_defined_tightness_if_dsc_rep_item_w_n_tightness_and_d_user_defined forces a representation that has a descriptive_representation_item with name N1 and description D1 to have another descriptive_representation_item with name N2. 'tightness': N1 - value of the .name attribute of the descriptive_representation_item; 'user defined': D1 - value of the .description attribute of the descriptive_representation_item; 'user defined tightness': N2 - value of the .name attribute of the other descriptive_representation_item.

EXPRESS specification:

```

RULE dsc_rep_item_w_n_user_defined_tightness_if_dsc_rep_item_w_n_tightness_
  and_d_user_defined FOR (REPRESENTATION);

  LOCAL
    rep_set:  SET OF representation := [];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

```

```

REPEAT i := 1 TO HIINDEX(REPRESENTATION) WHILE NOT violation;
    violation := (SIZEOF(QUERY(r <* REPRESENTATION[i].ITEMS |
        (r.NAME = 'tightness') AND (r.DESRIPTION = 'user defined'))
        > 0) AND
        (SIZEOF(QUERY(r <* REPRESENTATION[i].ITEMS |
            (r.NAME = 'user defined tightness')) = 0);
END_REPEAT;

WHERE
    WR1: NOT violation;
END_RULE;

```

5.2.5.290 same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_structural_system_symmetry_relationship

The same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_structural_system_symmetry_relationship forces the relating and the related side of a relationship of type T_relationship to be of the same class if T_relationship is of class C. PRODUCT_DEFINITION: T - type that is related by T_relationship; 'structural system symmetry relationship': C - class of T_relationship.

EXPRESS specification:

```

RULE same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_
    structural_system_symmetry_relationship FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL
    cl_ass:      SET OF CLASSIFICATION_ASSIGNMENT := [];
    clfied_inst: SET OF PRODUCT_DEFINITION_RELATIONSHIP := [];
END_LOCAL;

cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'structural system symmetry relationship');

REPEAT i := 1 TO HIINDEX(cl_ass);
    clfied_inst := clfied_inst + cl_ass[i].items;
END_REPEAT;

WHERE
    WR1: SIZEOF(WHICH_CLASS(i.RELATING_PRODUCT_DEFINITION) -
        WHICH_CLASS(i.RELATED_PRODUCT_DEFINITION)) = 0;
END_RULE;

```

5.2.5.291 optional_occurrence_of_attributes_for_corrugated_structure_- boundary_relationship_design_parameters

The optional_occurrence_of_attributes_for_corrugated_structure_boundary_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'corrugated structure boundary relationship design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_corrugated_structure_boundary_
relationship_design_parameters FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['displacement'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'corrugated structure boundary relationship
design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.292 PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_- structure_design_definition_referenced_by_exactly_1_SHAPE_- ASPECT_w_class_id_border_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE gets all classification_assignment instances with id 'corrugated structure design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'border' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'corrugated structure design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
    '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'border' IN WHICH_CLASS(i))) < 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.293 PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_structure_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_structure_design_parameters gets all classification_assignment instances with id 'corrugated structure design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "corrugated structure design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_
  definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_
  DEFINITION_w_NAME_eq_corrugated_structure_design_parameters
  FOR (PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'corrugated structure design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'corrugated structure design parameters')) = 1);
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.294 PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_- structure_design_definition_referenced_by_exactly_1_SHAPE_- ASPECT_w_class_id_moulded_surface_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE gets all classification_assignment instances with id 'corrugated structure design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'moulded surface' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_structure_design_
  definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_
  via_OF_SHAPE
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'corrugated structure design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
    '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'moulded surface' IN WHICH_CLASS(i))) > 1);
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.295 one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_- design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'corrugated structure design parameters': VAL - value of the .name attribute of the property_definition_representation; 'thickness', 'tightness': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['thickness', 'tightness'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'corrugated structure design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.296 optional_occurrence_of_attributes_for_corrugated_structure_- design_parameters

The optional_occurrence_of_attributes_for_corrugated_structure_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'corrugated structure design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'corrugation width', 'user defined tightness': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_corrugated_structure_design_
parameters FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['corrugation width', 'user defined tightness'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'corrugated structure design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```


5.2.5.297 PROPERTY_DEFINITION_w_class_id_corrugated_structure_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_structure_function_parameters

The PROPERTY_DEFINITION_w_class_id_corrugated_structure_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_corrugated_structure_function_parameters gets all classification_assignment instances with id 'corrugated structure functional definition' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "corrugated structure function parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_corrugated_structure_functional_definition_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_corrugated_structure_function_parameters
FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'corrugated structure functional definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'corrugated structure function parameters')) = 1);
END_REPEAT;

WHERE

```

```

    wr1: NOT violation;
END_RULE;

```

5.2.5.298 optional_occurrence_of_attributes_for_corrugated_structure_- function_parameters

The optional_occurrence_of_attributes_for_corrugated_structure_function_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'corrugated structure function parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'user def function': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_corrugated_structure_function_
parameters FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['user def function'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'corrugated structure function parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.299 one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_function_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_function_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'corrugated structure function parameters': VAL - value of the .name attribute of the property_definition_representation; 'function': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_structure_function_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['function'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'corrugated structure function parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.300 optional_occurrence_of_attributes_for_panel_system_boundary_relationship_design_parameters

The optional_occurrence_of_attributes_for_panel_system_boundary_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL 'panel system boundary relationship design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_panel_system_boundary_relationship_
  design_parameters FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['displacement'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'panel system boundary relationship
  design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;
```

5.2.5.301 SHAPE_ASPECT_w_class_id_panel_system_curve_boundary_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_-system_curve_boundary_design_parameters

The SHAPE_ASPECT_w_class_id_panel_system_curve_boundary_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_-system_curve_boundary_design_parameters gets all classification_assignment instances with id 'panel system curve boundary' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "panel system curve boundary design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_panel_system_curve_boundary_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  panel_system_curve_boundary_design_parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'panel system curve boundary');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'panel system curve boundary design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;

```

END_RULE;

5.2.5.302 optional_occurrence_of_attributes_for_panel_system_curve_- boundary_design_parameters

The optional_occurrence_of_attributes_for_panel_system_curve_boundary_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'panel system curve boundary design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_panel_system_curve_boundary_
  design_parameters FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['displacement'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'panel system curve boundary design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j]))) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;
```

5.2.5.303 one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_curve- boundary_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_curve_boundary_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'panel system curve boundary design parameters': VAL - value of the .name attribute of the property_definition_representation; 'curve': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_curve_boundary_
  design_parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['curve'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'panel system curve boundary design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.304 one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'panel system design parameters': VAL - value of the .name attribute of the property_definition_representation; 'material offset', 'thickness', 'tightness': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['material offset', 'thickness', 'tightness'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'panel system design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```


5.2.5.305 optional_occurrence_of_attributes_for_panel_system_design_parameters

The optional_occurrence_of_attributes_for_panel_system_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'panel system design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'stiffener spacing', 'user defined tightness': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_panel_system_design_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['stiffener spacing', 'user defined tightness'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'panel system design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
      found := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.306 PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_- design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_- class_id_border_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE gets all classification_assignment instances with id 'panel system design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'border' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_
referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE
FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'panel system design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'SHAPE_ASPECT' + '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'border' IN WHICH_CLASS(i))) < 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.307 PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_- design_definition_referenced_by_exactly_1_PROPERTY_- DEFINITION_REPRESENTATION_via_DEFINITION_w_- NAME_eq_panel_system_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_design_parameters gets all classification_assignment instances with id 'panel system design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "panel system design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_panel_system_design_parameters
FOR(PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'panel system design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'panel system design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;

```

END_RULE;

5.2.5.308 PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_- design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_- class_id_moulded_surface_via_OF_SHAPE

The PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE gets all classification_assignment instances with id 'panel system design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the SHAPE_ASPECT instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via OF_SHAPE, filter out those SHAPE_ASPECT instances whose class id is 'moulded surface' check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_panel_system_design_definition_referenced_
  by_exactly_1_SHAPE_ASPECT_w_class_id_moulded_surface_via_OF_SHAPE
  FOR(PRODUCT_DEFINITION_SHAPE, SHAPE_ASPECT, APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF SHAPE_ASPECT := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
    'panel system design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'SHAPE_ASPECT' +
    '.' + 'OF_SHAPE');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    'moulded surface' IN WHICH_CLASS(i))) > 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.309 PROPERTY_DEFINITION_w_class_id_panel_system_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_function_parameters

The PROPERTY_DEFINITION_w_class_id_panel_system_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_system_function_parameters gets all classification_assignment instances with id 'panel system functional definition' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "panel system function parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_panel_system_functional_definition_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_panel_system_function_parameters
FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'panel system functional definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'panel system function parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.310 one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_- function_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_function_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'panel system function parameters': VAL - value of the .name attribute of the property_definition_representation; 'function': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_function_parameters
    FOR (representation);

    LOCAL
        reps:      BAG OF REPRESENTATION := [];
        arg_list:  LIST OF STRING := ['function'];
        violation:  BOOLEAN := FALSE;
    END_LOCAL;

    reps := QUERY(
        temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
                'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'panel system function parameters')
            )
        ) > 0
    );

    REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
        REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
            violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                rep_item.name = arg_list[j])) <> 1);
        END_REPEAT;
    END_REPEAT;

    WHERE
        wr1: NOT violation;

```

END_RULE;

5.2.5.311 optional_occurrence_of_attributes_for_panel_system_function_parameters

The optional_occurrence_of_attributes_for_panel_system_function_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'panel system function parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'user def function': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_panel_system_function_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['user def function'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
            'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'panel system function parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
      found := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.312 SHAPE_ASPECT_w_class_id_panel_system_plane_boundary_-referenced_by_exactly_1_PROPERTY_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_-system_plane_boundary_design_parameters

The SHAPE_ASPECT_w_class_id_panel_system_plane_boundary_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_panel_-system_plane_boundary_design_parameters gets all classification_assignment instances with id 'panel system plane boundary' for all instances of SHAPE_ASPECT in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a SHAPE_ASPECT instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "panel system plane boundary design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE SHAPE_ASPECT_w_class_id_panel_system_plane_boundary_referenced_by_
  exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  panel_system_plane_boundary_design_parameters
  FOR(SHAPE_ASPECT, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF SHAPE_ASPECT := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'panel system plane boundary');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'panel system plane boundary design parameters')) = 1);
END_REPEAT;

WHERE

```



```

    wr1: NOT violation;
END_RULE;

```

5.2.5.313 one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_plane- boundary_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_plane_boundary_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'panel system plane boundary design parameters': VAL - value of the .name attribute of the property_definition_representation; 'plane': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_panel_system_plane_boundary_
design_parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['plane'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'panel system plane boundary design parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.314 **PROPERTY_DEFINITION_w_class_id_plate_strike_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_strike_function_parameters**

The PROPERTY_DEFINITION_w_class_id_plate_strike_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_strike_function_parameters gets all classification_assignment instances with id 'plate strike functional definition' for all instances of PROPERTY_DEFINITION in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PROPERTY_DEFINITION instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "plate strike function parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PROPERTY_DEFINITION_w_class_id_plate_strike_functional_definition_
referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
w_NAME_eq_plate_strike_function_parameters
FOR(PROPERTY_DEFINITION, PROPERTY_DEFINITION_REPRESENTATION,
APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PROPERTY_DEFINITION := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'plate strike functional definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'plate strike function parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.315 one_item_in_rep_if_used_in_pdr_w_name_eq_plate_stroke_function_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_plate_stroke_function_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one `representation_item` whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a `property_definition_representation` whose 'name' attribute has a value VAL. 'plate stroke function parameters': VAL - value of the .name attribute of the `property_definition_representation`; 'function': arg_list - list of names for the .name attributes of the `representation_items` that shall belong exactly once to that representation:

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_plate_stroke_function_parameters
    FOR (representation);

    LOCAL
        reps:      BAG OF REPRESENTATION := [];
        arg_list:  LIST OF STRING := ['function'];
        violation: BOOLEAN := FALSE;
    END_LOCAL;

    reps := QUERY(
        temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
                'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'plate stroke function parameters')
            )
        ) > 0
    );

    REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
        REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
            violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                rep_item.name = arg_list[j])) <> 1);
        END_REPEAT;
    END_REPEAT;

    WHERE
        wr1: NOT violation;

```

END_RULE;

5.2.5.316 optional_occurrence_of_attributes_for_plate_strike_function_parameters

The optional_occurrence_of_attributes_for_plate_strike_function_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL: 'plate strike function parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'user def function': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation:

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_plate_strike_function_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['user def function'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'plate strike function parameters')
        )
      ) > 0
    );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
      found := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.317 user_def_class_assignments

The user_def_class_assignments forces an instance of type T that has an applied_classification_assignment with role.name = R and group.name = ID1 to have another applied_classification_assignment with role.name = R but group.name = ID2: product_definition: T - classified type 'class membership': R - role.name of the classification_assignments 'user defined': ID1 - class id (group.name) of the independent classification_assignment 'user defined class': ID2 - class id (group.name) of the dependent classification_assignment:

EXPRESS specification:

```

RULE user_def_class_assignments FOR (T, CLASSIFICATION_ASSIGNMENT);

LOCAL
    cl_ass:      SET OF CLASSIFICATION_ASSIGNMENT := [];
    clified_inst: SET OF GENERIC := [];
    violation:    BOOLEAN := FALSE;
END_LOCAL;

cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
                (i.ROLE.NAME = 'class membership')
                AND (i.ASSIGNED_CLASSIFICATION.NAME = 'user defined'));

REPEAT i := 1 TO HIINDEX(cl_ass);
    clified_inst := clified_inst + cl_ass[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(clified_inst) WHILE NOT violation;
    violation := NOT('user defined class' IN WHICH_CLASS(clified_inst[i]));
END_REPEAT;

WHERE
    WR1: NOT violation;
END_RULE;

```

**5.2.5.318 optional_occurrence_of_attributes_for_structural_system_-
adjacency_relationship_design_parameters**

The optional_occurrence_of_attributes_for_structural_system_adjacency_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL: 'structural system adjacency relationship design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'displacement': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation:

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_structural_system_adjacency_
relationship_design_parameters FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['displacement'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name =
    'structural system adjacency relationship design parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.319 one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_- design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'structural system design parameters': VAL - value of the .name attribute of the property_definition_representation; 'thightness': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['tightness'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'structural system design parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.320 optional_occurrence_of_attributes_for_structural_system_design_ parameters

The optional_occurrence_of_attributes_for_structural_system_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value VAL. 'structural system design parameters': VAL - value of the .name attribute of the property_definition_representation that uses the representation; 'user def tightness': arg_list - list of names for the .name attributes of the representation_items that shall belong zero or once to that representation.

EXPRESS specification:

```

RULE optional\_occurrence\_of\_attributes\_for\_structural\_system\_design\_parameters
  FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['user def  tightness'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'structural system design parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.321 one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_- symmetry_relationship_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_symmetry_relationship_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL. 'structural system symmetry relationship design parameters': VAL - value of the .name attribute of the property_definition_representation; 'mirroring plane': arg_list - list of names for the .name attributes of the representation_items that shall belong exactly once to that representation.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_structural_system_symmetry_
relationship_design_parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['mirroring plane'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
| (temp_prop_def_rep.name = 'structural system symmetry relationship
    design parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.322 mandatory_product_definition_plate_in_product_definition_panel_ system

The mandatory_product_definition_plate_in_product_definition_panel_system The product_
definition of class C_PD1 that also is a group (group_assignment.role.name = 'equiva-
lence') (typically an item_structure in the ARM) shall - if not empty - via the group
reference at least one product_definition of class C_PD2. 'panel system': C_PD1 - the
class of a product_definition that is linked to a group; 'plate': C_PD2 - the class of an-
other product_definition at least one of which shall be referenced by 'panel system'.

EXPRESS specification:

```

RULE mandatory_product_definition_plate_in_product_definition_panel_system

```

```

FOR (CLASSIFICATION_ASSIGNMENT, GROUP_ASSIGNMENT);

LOCAL
  cl_ass:    SET OF CLASSIFICATION_ASSIGNMENT := [];
  pd:        SET OF PRODUCT_DEFINITION := [];
  gr_ass:    SET OF GROUP_ASSIGNMENT := [];
  groups:    SET OF GROUP := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

cl_ass := QUERY(i <* CLASSIFICATION_ASSIGNMENT |
                (i.role.name = 'class membership') AND
                (i.assigned_classification.name = 'panel system'));

REPEAT i := 1 TO HIINDEX(cl_ass);
  pd := pd + i.items;
END_REPEAT;

gr_ass := QUERY(i <* GROUP_ASSIGNMENT |
                i.role.name = 'equivalence');

REPEAT i := 1 TO HIINDEX(gr_ass);
  REPEAT j := 1 TO HIINDEX(gr_ass[i].items);
    IF (gr_ass[i].items[j] IN pd) THEN
      groups := groups + gr_ass.assigned_group;
    END_IF;
  END_REPEAT;
END_REPEAT;

gr_ass := QUERY(i <* GROUP_ASSIGNMENT |
                (SIZEOF(i.items) > 0) AND
                (i.role.name = 'item structure') AND
                (i.assigned_group IN groups));

REPEAT i := 1 TO HIINDEX(gr_ass) WHILE NOT violation;
  violation := SIZEOF(QUERY(inst <* gr_ass[i].items |
                            'plate' IN WHICH_CLASS(inst))) = 0;
END_REPEAT;

WHERE
  WR1: NOT violation;
END_RULE;

```

5.2.5.323 PRODUCT_DEFINITION_SHAPE_w_class_id_structural_system_- design_definition_referenced_by_exactly_1_PROPERTY_- DEFINITION_REPRESENTATION_via_DEFINITION_w_- NAME_eq_structural_system_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_structural_system_design_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_-

w_NAME_eq_structural_system_design_parameters gets all classification_assignment instances with id 'structural system design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PROPERTY_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PROPERTY_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "structural system design parameters" check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_structural_system_design_definition_
  referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_
  w_NAME_eq_structural_system_design_parameters
  FOR (PRODUCT_DEFINITION_SHAPE, PROPERTY_DEFINITION_REPRESENTATION,
  APPLIED_CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PROPERTY_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME =
  'structural system design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PROPERTY_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set |
    i.NAME = 'structural system design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.324 max_one_version_id_per_versionable_object

The max_one_version_id_per_versionable_object ensures that a type that is referenced by an applied_identification_assignment whose role is 'version id' has exactly one reference of this type.

EXPRESS specification:

```

RULE max_one_version_id_per_versionable_object FOR(APPLIED_IDENTIFICATION_
  ASSIGNMENT);

LOCAL
  version_ids:      SET OF APPLIED_IDENTIFICATION_ASSIGNMENT := [];
  versionable_objects: BAG OF GENERIC := [];
  duplicate:        BOOLEAN := FALSE;
END_LOCAL;

version_ids := QUERY(i <* APPLIED_IDENTIFICATION_ASSIGNMENT |
  i.ROLE.NAME = 'version identifier');

REPEAT i := 1 TO HIINDEX(version_ids);
  versionable_objects := versionable_objects + version_ids[i].items;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
  REPEAT j := i + 1 TO HIINDEX(versionable_objects) WHILE NOT duplicate;
    duplicate := versionable_objects[i] := versionable_objects[j];
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT duplicate;
END_RULE;

```

5.2.5.325 PRODUCT_DEFINITION_SHAPE_w_class_id_beveled_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_beveled_groove_weld_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_beveled_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_beveled_groove_weld_design_parameters gets all classification_assignment instances with id 'beveled groove weld', for all instances of PRODUCT_DEFINITION_SHAPE in t1_set: get the PRODUCT_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PRODUCT_DEFINITION_REPRESENTATION instances whose attribute NAME has the value

”beveled groove weld design parameters” check if their number equals 1.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_beveled_groove_weld_referenced_by_
  exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_beveled_
  groove_weld_design_parameters
  FOR (PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'beveled groove weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'beveled groove
  weld design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.326 one_item_in_rep_if_used_in_pdr_w_name_eq_beveled_groove_weld_- design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_beveled_groove_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'beveled_groove_weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_beveled_groove_weld_design_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['endcut shape', 'taper', 'sideness',
                                'weld joint spacer', 'configuration', 'penetration',
                                'joint_orientation', 'tightness'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'beveled groove
                                weld design parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.327 ID_of_REPRESENTATION_with_class_definition_pattern

The ID_of_REPRESENTATION_with_class_definition_pattern forces string type attribute A of type T to match the pattern P if the instance of T is of class C.

EXPRESS specification:

```

RULE ID_of_REPRESENTATION_with_class_definition_pattern
FOR (CLASSIFICATION_ASSIGNMENT);

LOCAL

```

```

    clified_inst : SET OF GENERIC := [];
END_LOCAL;

REPEAT i := 1 TO HIINDX(CLASSIFICATION_ASSIGNMENT);
    IF (CLASSIFICATION_ASSIGNMENT[i].assigned_classification.name =
        'definition') THEN
        clified_inst := clified_inst + CLASSIFICATION_ASSIGNMENT[i].items;
    END_IF;
END_REPEAT;

WHERE
    WR1: SIZEOF(QUERY(i <* clified_inst | NOT(i.ID LIKE *:))) = 0;
END_RULE;

```

5.2.5.328 **PRODUCT_DEFINITION_SHAPE_w_class_id_butt_groove_weld_-referenced_by_exactly_1_PRODUCT_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_butt_-groove_weld_design_parameters**

The PRODUCT_DEFINITION_SHAPE_w_class_id_butt_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_butt_groove_weld_design_parameters gets all classification_assignment instances with id 'butt groove weld' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_butt_groove_weld_referenced_by_exactly_
1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_butt_groove_weld_
design_parameters
FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'butt groove weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

```

```

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PRODUCT_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME =
    'butt groove weld design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.329 one_item_in_rep_if_used_in_pdr_w_name_eq_butt_groove_weld_- design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_butt_groove_weld_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'butt_groove_weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_butt_groove_weld_design_parameters
  FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['face shape', 'sideness', 'weld joint spacer',
    'configuration', 'penetration', 'joint_orientation', 'tightness'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
        'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'butt groove weld design
          parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |

```



```

        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.330 PRODUCT_DEFINITION_SHAPE_w_class_id_fillet_weld_-referenced_by_exactly_1_PRODUCT_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_fillet_-weld_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_fillet_-weld_design_parameters gets all classification_assignment instances with id 'fillet weld' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_fillet_weld_referenced_by_exactly_1_
PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_fillet_-weld_
design_parameters
FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                i.ASSIGNED_CLASSIFICATION.NAME = 'fillet weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'fillet weld
design parameters')) = 1);

```

```

END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.331 PRODUCT_DEFINITION_SHAPE_w_class_id_continuous_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_continuous_fillet_weld_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_continuous_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_continuous_fillet_weld_design_parameters gest all classification_assignment instances with id 'continuous fillet weld' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_continuous_fillet_weld_referenced_by_
    exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
    continuous_fillet_weld_design_parameters
    FOR (PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
    CLASSIFICATION_ASSIGNMENT);

LOCAL
    c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
    t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
    t2_set: SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
    violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
    i.ASSIGNED_CLASSIFICATION.NAME = 'continuous fillet weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
    REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
        t1_set := t1_set + c_a_set[i].items[j];
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
    t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
    REPRESENTATION' + '.' + 'DEFINITION');
    violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'continuous fillet
    weld design parameters')) = 1);
END_REPEAT;

```

```

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.332 one_item_in_rep_if_used_in_pdr_w_name_eq_electrode_chemical_composition_design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_electrode_chemical_composition_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'electrode_chemical_composition_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_electrode_chemical_composition_
    design_parameters FOR (representation);

```

```

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['carbon', 'silicon', 'manganese',
        'phosphorus', 'sulphur', 'nicke', 'molydenum', 'aluminium',
        'titanium', 'zirconium', 'chromium', 'copper', 'vanadium'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name = 'electrode chemical
                composition design parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
            rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;

```

END_RULE;

5.2.5.333 one_item_in_rep_if_used_in_pdr_w_name_eq_fillet_weld_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_fillet_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'fillet_weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_fillet_weld_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['endcut shape type', 'sideness',
                                'configuration', 'penetration',
                                'joint_orientation', 'tightness'];

    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'fillet weld design parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;
```

5.2.5.334 PRODUCT_DEFINITION_SHAPE_w_class_id_groove_weld_-referenced_by_exactly_1_PRODUCT_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_groove_-weld_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_groove_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_groove_weld_design_parameters gets all classification_assignment instances with id 'groove weld' and gets the PRODUCT_DEFINITION_REPRESENTATION instances that are referencing a PRODUCT_DEFINITION_SHAPE instance via DEFINITION, filter out those PRODUCT_DEFINITION_REPRESENTATION instances whose attribute NAME has the value "groove weld design parameters".

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_groove_weld_referenced_by_exactly_1_
  PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_groove_weld_design_
  parameters
  FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'groove weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' +
    'PRODUCT_DEFINITION_REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'groove
    weld design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.335 one_item_in_rep_if_used_in_pdr_w_name_eq_groove_weld_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_groove_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'groove_weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_groove_weld_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['sideness', 'weld joint spacer',
                                'configuration', 'penetration', 'joint_orientation', 'tightness'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'groove weld design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.336 PRODUCT_DEFINITION_SHAPE_w_class_id_intermittent_fillet_- weld_referenced_by_exactly_1_PRODUCT_DEFINITION_- REPRESENTATION_via_DEFINITION_w_NAME_eq_- intermittent_fillet_weld_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_intermittent_fillet_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_intermittent_fillet_weld_design_parameters gets all classification_assignment instances with id 'intermittent fillet weld' and all instances of T1 that have class id 'intermittent fillet weld' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_intermittent_fillet_weld_referenced_by_
  exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_
  intermittent_fillet_weld_design_parameters
  FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'intermittent fillet weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
    REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'intermittent fillet
    weld design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.337 one_item_in_rep_if_used_in_pdr_w_name_eq_intermittent_fillet_- weld_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_intermittent_fillet_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'intermittent_fillet_weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_intermittent_fillet_weld_design_
parameters FOR (representation);

LOCAL
  reps:      BAG OF REPRESENTATION := [];
  arg_list:  LIST OF STRING := ['end rules', 'cutout rules',
                                'fillet alignment', 'fillet weld length', 'fillet weld spacing',
                                'penetration rules', 'endcut shape type', 'sideness',
                                'configuration', 'penetration', 'joint_orientation', 'tightness'];
  violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'intermittent fillet weld
          design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
    violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name = arg_list[j])) <> 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```


5.2.5.338 PRODUCT_DEFINITION_SHAPE_w_class_id_spot_seam_weld_-referenced_by_exactly_1_PRODUCT_DEFINITION_-REPRESENTATION_via_DEFINITION_w_NAME_eq_spot_-seam_weld_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_spot_seam_weld_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_spot_seam_weld_design_parameters gets all classification_assignment instances with id 'spot seam weld' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_spot_seam_weld_referenced_by_exactly_1_
  PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_spot_seam_weld_
  design_parameters
  FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'spot seam weld');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'spot seam weld
  design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.339 one_item_in_rep_if_used_in_pdr_w_name_eq_spot_seam_weld_- design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_spot_seam_weld_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'spot_seam_weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_spot_seam_weld_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['weld context', 'configuration',
                                   'penetration', 'joint_orientation',
                                   'tightness'];

    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'spot seam weld design
            parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.340 one_item_in_rep_if_used_in_pdr_w_name_eq_weld_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'weld_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_weld_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['shape of weld surface', 'weld geometry',
                                   'connection angle', 'rotation angle', 'inclination angle',
                                   'shrinkage', 'weld size'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'weld design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.341 **PRODUCT_DEFINITION_SHAPE_w_class_id_weld_filler_- material_referenced_by_exactly_1_PRODUCT_DEFINITION_- REPRESENTATION_via_DEFINITION_w_NAME_eq_weld_filler_- material__design_parameters**

The PRODUCT_DEFINITION_SHAPE_w_class_id_weld_filler_material_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weld_filler_material__design_parameters gets all classification_assignment instances with id 'weld filler material' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_weld_filler_material_referenced_by_
  exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_weld_
  filler_material__design_parameters
  FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
  CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'weld filler material');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'weld filler material
  design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.342 optional_occurrence_of_attributes_for_weld_filler_material_design_parameters

The optional_occurrence_of_attributes_for_weld_filler_material_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value 'weld_filler_material_design_parameters'.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_weld_filler_material_design_
parameters FOR (representation);

LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['chemical composition id', 'electrode id',
                                'material strength id', 'notch impact work id'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        SIZEOF (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'weld filler material design
                    parameters')
            )
        ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.343 PRODUCT_DEFINITION_SHAPE_w_class_id_welded_joint_- design_definition_referenced_by_exactly_1_PRODUCT_- DEFINITION_REPRESENTATION_via_DEFINITION_w_- NAME_eq_welded_joint_design_parameters

The PRODUCT_DEFINITION_SHAPE_w_class_id_welded_joint_design_definition_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_welded_joint_design_parameters gets all classification_assignment instances with id 'welded joint design definition' for all instances of PRODUCT_DEFINITION_SHAPE in t1_set.

EXPRESS specification:

```

RULE PRODUCT_DEFINITION_SHAPE_w_class_id_welded_joint_design_definition_
referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_
NAME_eq_welded_joint_design_parameters
FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set: SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set: SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
  i.ASSIGNED_CLASSIFICATION.NAME = 'welded joint design
  definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'welded joint
  design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.344 one_item_in_rep_if_used_in_pdr_w_name_eq_welded_joint_design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_welded_joint_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'welded_joint_design_parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_welded_joint_design_parameters
  FOR (representation);

  LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['configuration', 'penetration',
                                'joint_orientation', 'tightness'];
    violation: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
      SIZEOF (
        QUERY(
          temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
          | (temp_prop_def_rep.name = 'welded joint design parameters')
        )
      ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
      violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                rep_item.name = arg_list[j]))) <> 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.345 optional_occurrence_of_attributes_for_weld_design_parameters

The optional_occurrence_of_attributes_for_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one repre-

sentation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value 'weld_design_parameters'.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_weld_design_parameters
  FOR (representation);

  LOCAL
    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['start suspension', 'end suspension',
                                'weld cross section area'];
    found: BOOLEAN := FALSE;
  END_LOCAL;

  reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'weld design parameters')
      )
    ) > 0
  );

  REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
      found := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name=arg_list[j]))) > 1);
    END_REPEAT;
  END_REPEAT;

  WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.346 optional_occurrence_of_attributes_for_groove_weld_design_parameters

The optional_occurrence_of_attributes_for_groove_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value 'groove weld design parameters'.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_groove_weld_design_parameters
  FOR (representation);

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['weld backing type'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
  SIZEOF (
    QUERY(
      temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
      | (temp_prop_def_rep.name = 'groove weld design parameters')
    )
  ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (SIZEOF(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j])) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.347 optional_occurrence_of_attributes_for_beveled_groove_weld_- design_parameters

The optional_occurrence_of_attributes_for_beveled_groove_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value 'beveled groove weld design parameters'.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_beveled_groove_weld_design_
  parameters FOR (representation);

```

```

LOCAL
  reps: BAG OF REPRESENTATION := [];
  arg_list: LIST OF STRING := ['weld backing type', 'taper angle'];
  found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
  temp_rep <* representation |
    sizeof (
      QUERY(
        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'beveled groove weld
          design parameters')
      )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
  REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
    found := (sizeof(QUERY(rep_item <* reps[i].items |
      rep_item.name=arg_list[j]))) > 1);
  END_REPEAT;
END_REPEAT;

WHERE
  wr1: NOT found;
END_RULE;

```

5.2.5.348 **PRODUCT_DEFINITION_SHAPE_w_class_id_structural_part_- joint_design_definition_referenced_by_exactly_1_PRODUCT_- DEFINITION_REPRESENTATION_via_DEFINITION_w_- NAME_eq_structural_part_joint_design_parameters**

The **PRODUCT_DEFINITION_SHAPE_w_class_id_structural_part_joint_design_definition_-referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_structural_part_joint_design_parameters** gets all classification assignment instances with id 'structural part joint design definition' for all instances of **PRODUCT_DEFINITION_SHAPE** in t1_set: get the **PRODUCT_DEFINITION_REPRESENTATION** instances that are referencing a **PRODUCT_DEFINITION_SHAPE** instance via **DEFINITION**, filter out those **PRODUCT_DEFINITION_REPRESENTATION** instances whose attribute **NAME** has the value "structural part joint design parameters" check if their number equals 1.

EXPRESS specification:

```
RULE PRODUCT_DEFINITION_SHAPE_w_class_id_structural_part_joint_design_
```

```

definition_referenced_by_exactly_1_PRODUCT_DEFINITION_REPRESENTATION_via_
DEFINITION_w_NAME_eq_structural_part_joint_design_parametes
FOR(PRODUCT_DEFINITION_SHAPE, PRODUCT_DEFINITION_REPRESENTATION, APPLIED_
CLASSIFICATION_ASSIGNMENT);

LOCAL
  c_a_set: SET OF CLASSIFICATION_ASSIGNMENT := [];
  t1_set:  SET OF PRODUCT_DEFINITION_SHAPE := [];
  t2_set:  SET OF PRODUCT_DEFINITION_REPRESENTATION := [];
  violation: BOOLEAN := FALSE;
END_LOCAL;

c_a_set := QUERY(i <* APPLIED_CLASSIFICATION_ASSIGNMENT |
                 i.ASSIGNED_CLASSIFICATION.NAME = 'structural part joint
                 design definition');

REPEAT i := 1 TO HIINDEX(c_a_set);
  REPEAT j := 1 TO HIINDEX(c_a_set[i].items);
    t1_set := t1_set + c_a_set[i].items[j];
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 TO HIINDEX(t1_set);
  t2_set := USEDIN(t1_set[i], 'SHIP_STRUCTURES.' + 'PRODUCT_DEFINITION_
  REPRESENTATION' + '.' + 'DEFINITION');
  violation := NOT (SIZEOF(QUERY(i <* t2_set | i.NAME = 'structural part
  joint design parameters')) = 1);
END_REPEAT;

WHERE
  wr1: NOT violation;
END_RULE;

```

5.2.5.349 optional_occurrence_of_attributes_for_butt_groove_weld_design_ parameters

The optional_occurrence_of_attributes_for_butt_groove_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') zero or one representation_item who's 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation who's 'name' attribute has a value 'butt groove weld design parameters'.

EXPRESS specification:

```

RULE optional_occurrence_of_attributes_for_butt_groove_weld_design_parameters
  FOR (representation);

  LOCAL

```

```

    reps: BAG OF REPRESENTATION := [];
    arg_list: LIST OF STRING := ['weld backing type'];
    found: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
        sizeof (
            QUERY(
                temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
                | (temp_prop_def_rep.name = 'butt groove weld design
                    parameters')
            )
        ) > 0
    );

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT found);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT found);
        found := (sizeof(QUERY(rep_item <* reps[i].items |
            rep_item.name=arg_list[j])) > 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT found;
END_RULE;

```

5.2.5.350 one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_joint_- design_parameters

The `one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_joint_design_parameters` forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value 'structural part joint design parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_structural_part_joint_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['tightness', 'joint orientation'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

```

```

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(
            temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
            | (temp_prop_def_rep.name = 'structural part joint design
              parameters')
        )
    ) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
                                   rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.5.351 one_item_in_rep_if_used_in_pdr_w_name_eq_continuous_fillet_weld_- design_parameters

The one_item_in_rep_if_used_in_pdr_w_name_eq_continuous_fillet_weld_design_parameters forces the 'items' attribute of a representation to have for each entry in the list ('...') exactly one representation_item whose 'name' attribute has the value given in the list if the representation is the 'used_representation' in a property_definition_representation whose 'name' attribute has a value VAL 'continuous fillet weld design parameters'.

EXPRESS specification:

```

RULE one_item_in_rep_if_used_in_pdr_w_name_eq_continuous_fillet_weld_design_
parameters FOR (representation);

LOCAL
    reps:      BAG OF REPRESENTATION := [];
    arg_list:  LIST OF STRING := ['endcut shape type', 'sideness',
                                   'configuration', 'penetration', 'joint_orientation', 'tightness'];
    violation: BOOLEAN := FALSE;
END_LOCAL;

reps := QUERY(
    temp_rep <* representation |
    SIZEOF (
        QUERY(

```

```

        temp_prop_def_rep <* USEDIN(temp_rep,
'SHIP_STRUCTURES.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')
        | (temp_prop_def_rep.name = 'continuous fillet weld design
        parameters')
    )
) > 0
);

REPEAT i:=1 TO HIINDEX(reps) WHILE (NOT violation);
    REPEAT j:=1 TO HIINDEX(arg_list) WHILE (NOT violation);
        violation := (SIZEOF(QUERY(rep_item <* reps[i].items |
        rep_item.name = arg_list[j])) <> 1);
    END_REPEAT;
END_REPEAT;

WHERE
    wr1: NOT violation;
END_RULE;

```

5.2.6 Ship structures function definitions

5.2.6.1 WHICH_CLASS

The function WHICH_CLASS determines the applied_classification_assignments pointing to an instance of an arbitrary type and returns the class ids in a string list.

EXPRESS specification:

```

FUNCTION WHICH_CLASS(T: GENERIC): LIST OF STRING;
    LOCAL
        elements: LIST OF APPLIED_CLASSIFICATION_ASSIGNMENT;
        class_list: LIST OF STRING :=[];
    END_LOCAL;

    elements :=
        USEDIN(T, 'SHIP_STRUCTURES.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS');

    REPEAT i:=1 TO HIINDEX(elements);
        IF (elements[i]\classification_assignment.role.name =
        'class membership') THEN
            class_list := class_list +
            elements[i]\classification_assignment.assigned_classification\group.name;
        END_IF;
    END_REPEAT;

    RETURN(class_list);
END_FUNCTION;

```

5.2.6.2 get_company_id

The function get_company_id extracts the company id that is a unique identifier for the company.

EXPRESS specification:

```

FUNCTION get_company_id(id: STRING): STRING;
  LOCAL
    company_id: STRING;
    i: INTEGER;
  END_LOCAL;

  CONSTANT
    sep: STRING := '.';
  END_CONSTANT;

  REPEAT i := 1 TO LENGTH(id) WHILE (id[i] <> sep);
    company_id := company_id + id[i];
  END_REPEAT;

  RETURN (company_id);
END_FUNCTION;

```

5.2.6.3 get_local_id

The function get_local_id extract the local id that uniquely identifies the localidentifier throughout the company.

EXPRESS specification:

```

FUNCTION get_local_id(id: STRING): STRING;
  LOCAL
    local_id: STRING;
    i: INTEGER;
    j: INTEGER;
  END_LOCAL;

  CONSTANT
    sep: STRING := '.';
  END_CONSTANT;

  j := 1;
  REPEAT i := 1 TO LENGTH(id) WHILE (id[i] <> sep);
    j := i + 1
  END_REPEAT;

```

```
REPEAT i := j TO LENGTH(id);  
    local_id := local_id + id[i];  
END_REPEAT;  
  
RETURN (local_id);  
END_FUNCTION;
```


6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods: ISO 10303-21, ISO 10303-22.

Requirements with respect to implementation methods-specific requirements are specified in annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combination of options that may be included in the implementation. The PICS proforma is provided in annex D.

This part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into the following conformance classes:

- Class 1, ship structures definition and approval data that is created at the preliminary design stage of a ship, has structural definitions and shape representations of this stage, shall be exchanged between the shipyard and the subcontractor; and, the early class approval data for the preliminary design of the ship, including the definition of hull cross sections, has class approvals with regard to the detailed design definitions, shall be exchanged between the shipyard and the classification society;
- Class 2, ship structures definition and approval data that is elaborated at the detailed design stage of a ship, under consideration of the production design of ship structures, shall be exchanged between the shipyard and the subcontractor;
- Class 3, ship structures definition and approval data that is completed at the product manufacturing stage of a ship, has manufacturing and welding definitions, shall be exchanged between the design department and the manufacturing department of the shipyard;
- Class 4, class approval data for the structural parts (plates and profiles) of the ship, has class approvals with regard to the manufacturing definitions, shall be exchanged between the shipyard and the classification society.

Support for a particular conformance class requires support of all the options specified in this class.

Table 1 – Conformance classes

Unit of functionality	Class 1 early design & class approval	Class 2 detailed design	Class 3 manufacturing class	Class 4 approval
	Subcontractor ⇔ Shipyard ⇔ Class. Society	Subcontractor ⇔ Shipyard	Design Dept. ⇔ Manuf. Dept.	Shipyard ⇔ Class. Society
class_approvals	X			X
configuration_management	X	X	X	X
definitions	X	X	X	X
design_loads	X	X		X
external_references	X	X	X	X
hull_cross_sections	X			
integrated_resources	X	X	X	X
items	X	X	X	X
location_concept	X	X	X	X
product_structures	X	X	X	X
representations	X	X	X	X
shapes	X	X	X	X
ship_general_characteristics	X	X	X	X
ship_manufacturing_definitions		X	X	X
ship_materials	X	X	X	X
ship_measures	X	X	X	X
structural_features		X	X	X
structural_parts		X	X	X
structural_systems	X	X	X	X
support_resources	X	X	X	X
welds		X	X	X

Annex A
(normative)

AIM EXPRESS expanded listing

To be added at a later date.

Annex B
(normative)

AIM short names

To be added at a later date.

Annex C (normative)

Implementation method specific requirements

The implementation method defines what types of exchange behaviour are required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and in the AIM defined in annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'ship_structures_schema'.

Annex D
(normative)

Protocol Implementation Conformance Statement (PICS)
proforma

To be added at a later date.

Annex E
(normative)

Information object registration

E.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(218) version(-1) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

E.2 Schema identification

To provide for unambiguous identification of the schema specifications given in this application protocol **ship_structures_schema** in an open information system, object identifiers are assigned as follows:

{ iso standard 10303 part(218) version(1) object(1) ship-structures-schema(1) }

is assigned to the **ship_structures_schema** expanded schema (see annex A).

The meaning of these values is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex F (informative)

Application activity model

The application activity model (AAM) is provided as an aid in understanding the scope and information requirements defined in this application protocol. The model is presented as a set of figures that contain the activity diagrams and a set of definitions of the activities and their data. The application activity model is given in 26. Activities and data flows that are out of scope are marked with an asterisk.

F.1 Application activity model definitions and abbreviations

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol.

The definitions given in this annex do not supersede the definitions given in the main body of the text.

F.1.1

Approve Design of Ship Structure (A242)

the top level activity for the approbation of ship design

NOTE It is used as the entry activity for both Design Approval Preview, Approval of primary design (primary structure) and approval of detail design (secondary structure). The ship is not certified by this activity alone, in addition it is needed to perform construction surveys and testing before certificates are issued.

F.1.2

alarm*

the document or report resulting by operating a ship that warns the maintenance personnel which problem the ship has

F.1.3

approved design

the final design to be submitted as an offer

F.1.4

approved primary design

the approved fundamental and principal structural design

NOTE It is a result from the approve design of ship structure activity by a classification society.

F.1.5

arrangements*

the ship's compartments and spaces

NOTE Any description of arrangements will include associated definitions of purpose for the compartment or space.

F.1.6

Assemble Ship (A33)

the production stage that assembles the modular units, the serviced parts and additional material which result from the production of a steel sub-section

NOTE The result is an assembled ship, that still has to be tested.

F.1.7

assembly definition

the rough division into assemblies

F.1.8

authorities

the governmental agency or corporation to administer a statutory and class survey

F.1.9

availability, reliability and maintainability information

the information about the components that is required to install them in the ship and is required for planned maintenance

F.1.10

basic hull parameters*

estimated principal dimensions based on historical data or preliminary design development

F.1.11

budget*

the cost constraint on the design building and maintenance of the ship

F.1.12

building capacities of yard

the production ability available to the yard, usually in tones per year

F.1.13

building sequences

the sequences of producing a ship, as input for preparing the section plans

F.1.14

Calculate Cost of Ship* (A124)

the description of creating the negotiating documents based on technical product data and the calculation of their estimated manufacturing cost

NOTE The results of this activity may contain sale price documents, financing support plan and documents describing funding and possible loans.

F.1.15

certificates*

the certificates issued by the Classification Society on completing the ship

F.1.16

Check Bottom Structure (A24241)

checking the hull structure within the area of the lowest part of the hull, including tanks, ballast area and keel to confirm conformance with the regulations

F.1.17

Check Bulkheads (A24244)

checking the dimensioning, the correct type and the correct position of the bulkheads

F.1.18

Check Deck Plans (A24243)

controlling whether the necessary spaces and volumes of the vessel are correct in volume, size and position

F.1.19

Check Design of General Arrangement* (A2422)

the arrangement checking

NOTE It is basically a checking of watertight integrity arrangements and stability conditions (intact and damage stability). The requirements are considered to meet the relevant regulations of the international Convention on load lines. 1969. Attention should, however, be given to possible additional requirements of the Maritime Authorities in the country in which the ship is to be registered. For passenger ships the "International Convention for Safety Of Life At Sea (SOLAS)" will normally apply.

F.1.20

Check Design of Girder Strength* (A2423)

checking the compliance with rule requirements with respect to global strength requirements

NOTE This approval is necessary before the yard can start ordering steel.

F.1.21

Check Global Strength and Secondary Structure (A2424)

checking the strength of the whole ship for the approval of the secondary design as part of the approval and certification process

NOTE The content of this activity is the same for all ships when it comes to conformance with main class rules, but varies when it comes to additional class rules (type of vessel) and register notations.

F.1.22**Check Side Structure (A24242)**

checking the sufficient material thickness and the fulfillment of the regulation requirements

F.1.23**Check WCoG (A2432)**

checking the weight and center of gravity for the design

F.1.24**classification society**

an organization that enhances the safety of life and property at sea by providing rules, regulations and personnel for assessing and classifying ships during their lifecycle

F.1.25**Collect and Analyze Design Transfer into Drawings* (A2441)**

the necessary activities to ensure the material supply for the production of the ship

F.1.26**Collect Part Order Information (A24344)**

collecting part ordering information to be used in management information systems (MIS) and estimations

F.1.27**collision control results**

the results from managing the control for clutches

NOTE The clutches are, both hard and soft, between different design parts, such as pipes, equipment items and cable trays.

F.1.28**Complete and Approve Design of Machinery* (A25)**

the selection, arrangement and approval of the power plant in terms of the main engine, associated propulsion system and its auxiliary machinery

F.1.29**Complete and Approve Design of Outfitting and Distribution Systems* (A26)**

the selection and approval of the necessary outfitting equipment

NOTE The selection is based mainly on former designs and in accordance with the requirements. It also contains the layout of the different types of distribution systems such as piping and HVAC.

F.1.30

Complete and Approve Design of Ship Structure (A24)

the completion and approval of the ship structural design

F.1.31

Complete and Approve Ship Design (A2)

the production and approval of ship design product data, documents and the classification drawings using the preliminary design from the bid preparation, as well as the required rules and regulations

NOTE The result of this activity is the approved design and the production and delivery schedule.

F.1.32

Conduct Contractor Sea Trials* (A343)

the sea trials to test if the built ship meets the contract requirements

F.1.33

Conduct Acceptance Trials* (A344)

the final trials for delivery the ship

F.1.34

configuration plan for outfitting

one part of the production documents for steel structures

F.1.35

contract*

the output from the activity which involves placing the order for the ship

NOTE The contract is used as a constraint in subsequent activities such as final design and approval and production.

F.1.36

Coordinate Subcontractors Tasks* (A24434)

the distribution and coordination of the tasks which are done by the subcontractors

F.1.37

cost*

the calculated cost of the ship based on the cost of material and labour

F.1.38

Create Design of Classification Items (A2412)

creating the design of those items required by the classification society

EXAMPLE 1 Midship sections, bulkheads, shell profile arrangement and foundations

F.1.39

Create Preliminary Design (A122)

all design activities relevant in a very preliminary stage of ship design in consideration of classification rules, national/international demands, shipyard constraints and owner requirements

NOTE The aim of this task is to make a shipyard offer.

F.1.40**Create Preliminary General Arrangements* (A1222)**

producing the preliminary compartmentation plans from the preliminary hull form definition

F.1.41**Create Preliminary Hull Form* (A1221)**

the first step of designing a ship, by which the hull form of a ship in a very preliminary stage of ship design shall be created

NOTE Using parent ships main dimensions and form parameters one or more preliminary hull forms will be generated.

F.1.42**Create Preliminary Machinery Design* (A1225)**

producing the preliminary designs for the ship machinery

NOTE It includes the prime mover, shaft system, fuel system, power systems and cargo handling equipment.

F.1.43**Create Preliminary Outfitting Design* (A1226)**

producing the preliminary design for the ship's outfitting

NOTE It includes distributed systems, such as piping and electrical systems.

F.1.44**Create Preliminary Structure Design (A1224)**

producing the preliminary steel structure design

NOTE It includes the arrangement of the primary structural members.

F.1.45**Create Production Documents (A2442)**

producing the production documents which are needed to describe the features of technical parts and their relationship

F.1.46

crew

the person who have duties on a ship

F.1.47

critical design areas

relating to the design areas at which a change may be expected when performing a design approval preview

F.1.48

cut-out definition

the definition of the cutouts when defining the notches, lag and welding of the plates

F.1.49

Decide Post-sales and Maintenance Support* (A123)

putting together the maintenance package for the ship

NOTE This is part of the tender document and includes the post sales support.

F.1.50

Decommission and Disassemble* (A5)

the functions of disassembly and preparing for reuse and recycling of parts and materials of the product ship

NOTE The activity covering all activities related to the last state of the ships life cycle.

F.1.51

Define Brackets (A243413)

add brackets to early and secondary designs

F.1.52

Define Endcuts, Bevelling and Welding (A243433)

defining the design features required for preparing structural parts for the welding of profiles

F.1.53

Define Holes and Notches (A243432)

fixing the dependent holes and notches in their position when defining the profiles in their length and position

F.1.54

Define Holes and Penetrations (A243422)

defining the holes and penetrations according to the other functions of the system that has the connected throughputs

F.1.55

Define Insulation and Surface Treatment (A24345)

defining the insulation and surface treatment definitions to be used depending on location of structure and functional requirements

F.1.56

Define Notches, Lag and Welding (A243423)

all design activities which are meant to support the welding process

F.1.57

Define Plates (A243411)

setting the layout of individual plates according to the structural hull design

F.1.58

Define Profiles (A243412)

setting the position of the profiles according to the structural hull and related design

F.1.59

Define Seams, Bevelling and Welding (A243421)

defining the designed features of structured parts for the welding of plates

F.1.60

design change request

the feedback from approval design of ship structure activity

NOTE It is the necessary change of the primary structural design of being requested.

F.1.61

Design Collision Bulkhead (A241212)

designing the collision bulkhead, specially concerning water resistance and strength in the bow area

F.1.62

Design Longitudinal, Horizontal Structures (A24122)

supporting the design of decks, bottoms and horizontal shell profiles

F.1.63

Design Longitudinal, Vertical Structures (A24123)

supporting the design of vertical structural separations such as bulkheads, cofferdams and vertical shell profiles

F.1.64

Design of Transverse Structures (A24121)

dimensioning the midship section, including plates, shell profiles and shell thickness

F.1.65

design schedule*

data that controls the time from the design phase to production

F.1.66

Design Secondary Profiles (A243434)

selecting the detailed length of the individual profiles and their size and cross-section

F.1.67

Design Secondary Structure (A24341)

designing structure details not defined in early design

F.1.68

Design Ship Structure (A241)

covering the activities of creating a synthesis of the product specification and transforming them into the physical representation of the product

NOTE In this case the function is understood to contain mainly the activities to satisfy the regulations set up by the classification society.

F.1.69

Design Special Areas (A24125)

design areas of the ship not covered by the longitudinal and vertical structure

EXAMPLE 1 The engine foundations as well as aft and fore body

F.1.70

Design Transverse Bulkheads (A241211)

finishing the design by completing the dimensioning of the transverse structure

F.1.71

Design Transverse Frames (A241213)

design frames between transverse bulkheads

NOTE The frames are the transverse members that make up the riblike part of the skeleton of a ship.

F.1.72

detail design

the detailed structural design with extended treatment of the primary structural design and attention to particular items, i.e. the structural parts

NOTE It is a result from the activity elaborate ship structure design

F.1.73

distribution and outfitting design*

the design of the distribution and outfitting systems

F.1.74**drawing**

the technical representations, outlining figures or plans of the ship structures

NOTE The drawing is the results of detail design and input for creating the production documents.

F.1.75**early design**

the design of the ship structures including the primary design and approved primary design

F.1.76**Elaborate Ship Structure Design (A243)**

the function giving detailed information to order parts and to manufacture, and giving the physical outcome of the design

NOTE Among these activities are the subdividing of physical structures and the determination of tasks for manufacturing.

F.1.77**engineering consultants**

organizations that provide specific engineering services to shipyards, ship owners and classification societies during the ship lifecycle

F.1.78**equipment certificates***

the certificates issued by the Classification Society on completing the equipment items which will be assembled to create the final product

F.1.79**Estimate Drydock Resources and its Division (A24431)**

the function that engages with the time scheduling and assignment of dry dock capacities

F.1.80**Estimate Hydrodynamics and Powering***

approximating hydrodynamic properties data calculations

EXAMPLE 1 Resistance, propulsion, seakeeping and maneuverability for the preliminary hull form.

F.1.81**Evaluate Request and Schedule Bid* (A121)**

shipyard activities when evaluating the inquiry of the ship owner for a new ship

F.1.82

feedback

the outputs from activities which then feed back and modify previous activities in the lifecycle on the current or subsequent ships

F.1.83

Finalize and Approve General Arrangements* (A21)

finalizing the details of the general arrangement after having created a draft layout

NOTE The ship's systems are described by a compartment and access drawing showing the location, the access, and the size of the different compartments.

F.1.84

Finalize and Approve Hull Form* (A22)

final checking the hull form from the preliminary design resulting in a final and approved hull form design

F.1.85

Finalize and Approve Hydrodynamics and Powering* (A23)

final checking the results of the all relevant hydrodynamic calculations

EXAMPLE 1 Resistance, propulsion, seakeeping and maneuverability.

F.1.86

Handle Plates (A24342)

splitting plate parts and add production information

F.1.87

Handle Profiles (A24343)

partition profiles, add production information and design additional and secondary profiles

F.1.88

historical data from previous designs

data held by the shipyard or model basin on previous ship designs and used to estimate the hydrodynamics, powering requirements and sea-keeping

F.1.89

hole definition

the geometrical and topological information of a hole when defining holes and penetrations on a plate

F.1.90

hull design

the design of hull form, results of the hull detail design

F.1.91

hull form sections

the design of the hull moulded form at planar sections taken along the longitudinal axis of the ship

F.1.92**hull moulded form**

the definition of the hull shape of the ship, resulting from the addition of aft-body, mid-body and fore-body definitions

NOTE The hull moulded form does not take into account the thickness of the material from which the hull is made.

F.1.93**hydrodynamical loads**

the loads caused by hydrodynamical forces

F.1.94**hydrodynamics and powering results***

the results of calculations and model basin tests

NOTE The results contain resistance, propulsion, propeller performance, brake power, service speed, sea keeping and maneuverability data.

F.1.95**inspection information***

the information resulting from visual, non-destructive examinations or tests carried out to assess the condition of one or more components of a ship's system

F.1.96**knowledge and experience**

the previous experience and knowledge of companies involved throughout the ship lifecycle

F.1.97**laws, rules and regulations**

national laws, statutory regulations and classification society rules that are used to control the design, manufacture, operation, maintenance and scrapping of the ship

F.1.98**Layout Primary Structure (A24124)**

the entry activity for design approval preview and the approval of the primary design

F.1.99**list of classification items**

the list of those items required by the classification society

EXAMPLE 1 Midship sections, bulkheads, shell profile arrangement and foundations

F.1.100

list of required certificates*

the list supplied by the owner for certificate requirements as a result of placing an order

F.1.101

loading and stability manual*

the booklet which is placed on board of the ship for the information of the master, which enables him or her to load the ship within prescribed limits, relating to strength and stability

F.1.102

loads

a mass, force or weight supported by a ship

F.1.103

longitudinal bulkheads walls

the bulkheads walls between transverse frames

F.1.104

machinery design*

the design drawings and electronic models of the ship mechanical systems

NOTE It is an output from the final design process.

F.1.105

main cross section, longitudinal profile dimension and plate dimensions

the dimensions of the main hull cross section and longitudinal profiles and plates resulted from the transverse structure design

F.1.106

Maintain a Ship* (A42)

the maintenance activities during the operation phase of a ship

NOTE It is based on the existence of a planned maintenance system (PMS) for preventive maintenance on board, at the harbor base and at a yard. Preventative maintenance includes planned regular inspections, lubricating, cleaning, replacement of parts and may be scheduled by calendar date, usage condition. A planned maintenance system contains schedule, procedures, and a listing of required materials, skill, tools and test equipment. Corrective maintenance (repair) will be applied in case of a failure of a system or component. It may be predictable but is obviously not scheduled. Generally after discovering the failure a failure analysis and a specific corrective maintenance plan, inclusive a repair procedure will be made. Execution of the maintenance is followed by inspection, testing, inspection and approval by an organization authorized for the specific maintenance.

F.1.107

maintenance history*

the diary of the maintenances of a ship

F.1.108

maintenance personnel*

the person who have duties for maintaining a ship

F.1.109

maintenance report*

the report on the maintenance results

F.1.110

maintenance test results*

the results of functional tests carried out after the execution of maintenance actions

F.1.111

Make Hull Detail Design (A2434)

covering the design and realization of the steel structure by the use of plates and as stiffening element the profiles

NOTE Therefore in a first step the secondary structure is defined. Finally the structure is coated and in compartmentalization areas an insulation is necessary to be designed.

F.1.112

Make Initial Assembly Definition (A2431)

the first rough division into assemblies

NOTE This is further developed during production design.

F.1.113

Make Outfitting Detail Design* (A2435)

develop further the outfitting design made at early stages, and design all systems and functionalities not done in early design

F.1.114

Make Production Design of Ship Structure (A244)

the function that contains all activities which perform the material supply, the creation of production information and its documentation and the refined working information

F.1.115

Manage Collision Control* (A2433)

checking the design for clutches

NOTE The clutches are, both hard and soft, between different design parts, such as pipes, equipment items and cable trays.

F.1.116

manufacturing restrictions

a constraint on the ship construction and design processes governed by available technology and shipyard facilities

F.1.117

material list

the list of raw materials needed to manufacture the ship

NOTE A result of the final design process.

F.1.118

modification

a feedback from the modification for classification when designing the ship structure

F.1.119

modifications from machinery*

modifications to the hydrodynamics and powering due to feedback from the preliminary machinery design

F.1.120

modifications to hull form*

modifications to the hull shape due to feedback from hydrodynamics and powering results and the final design process

F.1.121

Modify for Classification* (A2413)

the function concerned with the items of classification found to be not solved properly according to the rules.

NOTE The inputs are the design change requests which are modified to be satisfactory for the technical solution.

F.1.122

modular units*

sub-sections of the ship complete with machinery and outfitting which will be assembled to create the final product

F.1.123

offer

containing the shipyard's data for producing the requested ship resulting from the preliminary design process

F.1.124

offer guidelines*

the data necessary to make an unconditional offer to the ship owner

F.1.125**operation manuals***

the booklets which are placed on board for the information, which enables the crew to operate the ship

F.1.126**operational information***

the document resulting by operating, maintaining and surveying a ship to give the information about the ship condition

F.1.127**Operate a Ship* (A41)**

provides feedback into other activities based on running the ship

NOTE The main aspect of ship operation in this model is with respect to maintenance. This maintenance process is controlled by "planned maintenance system" which is performed during the "create ship design" activity, and further more controlled by classification rules.

F.1.128**Operate and Maintain a Ship* (A4)**

running and maintenance of the ship during its service lifetime

F.1.129**operational histories***

the diary made during the operation of s ship

F.1.130**operational information***

accumulated information during the operation phase of the ship used for maintenance and in the final scrapping stage

F.1.131**owner**

the organization which requests, orders, takes delivery of and, for the purposes of this model, operates the ship

F.1.132**owner request and requirements**

the requirements document that is submitted to the shipyard by the owner upon the invitation to tender a bid

F.1.133**part ordering information**

the information to order the parts

NOTE It is obtained from the hull detail design. After that the plates and profiles have been

handled and the parts are collected.

F.1.134

Partition Profiles (A243431)

defining the resulting length of plate related stiffeners

F.1.135

Perform Design Approval Preview (A2421)

approval preview of ship design

NOTE This activity is very much a feasibility study where the design is checked very roughly to detect critical areas for thorough investigation and conformity checking both as a design comment and to draw attention to specific areas.

NOTE During design approval the content of this activity may vary with contract specifications.

F.1.136

Perform Ship Life Cycle (A0)

all of the lifecycle activities associated with a ship

F.1.137

Place Order* (A13)

an activity by owner that places an order for a ship from the bids that have been submitted

NOTE From this a contract is awarded.

F.1.138

planned maintenance system*

data created during the final design process and used during the operation and maintenance of the ship

F.1.139

plate info

the result from the defining plates during the secondary structure design state

NOTE It includes material type, thickness and geometrical form of the plates.

F.1.140

pre layout

the very initial layout of the ship which is produced during the bid evaluation stage and is the basis for the preliminary design

F.1.141

preliminary design

the design which is completed in the phases leading up to the submission of the tender

F.1.142

preliminary general arrangements*

the definition of the ship general arrangements as a result of the preliminary design process

F.1.143

preliminary hull form*

the definition of the hull form as a result of the preliminary design process

NOTE It is used in the offer documents and for preliminary compartment design, hydrodynamics and powering calculations.

F.1.144

preliminary machinery design*

the definition of the ship mechanical systems

NOTE Used early to estimate the noise, speed and vibration and to estimate the machinery weights.

F.1.145

preliminary machinery, structure and outfitting design

feedback consisting of the preliminary designs for machinery, structure and outfitting and furnishing

NOTE This allows the creation of preliminary general arrangements.

F.1.146

preliminary outfitting design*

the definition of the ship's outfitting and accommodation

NOTE It is resulted from the preliminary design process.

F.1.147

preliminary structure design

the definition of the preliminary ship structure during the preliminary design process

F.1.148

Prepare Bid (A12)

all activities of the shipyard regarding preparation and submission of the offer to the ship owner for the ship to be built

F.1.149

Prepare Outfitting and Equipping Plans and Schedules* (A24433)

work out outfitting and equipping plans

F.1.150

Prepare Production Documents for Steel Structure (A24421)

the function that performs the segmentation of the steel structure, creating working and assembly information

F.1.151

Prepare Section Plans (A24432)

the function that coordinates the proper sequence of the steel structure and outfitting of the structure

F.1.152

Present Offer* (A125)

the activity concerned with presentation of the offer to build the ship to the prospective ship owner

F.1.153

primary design

the fundamental and principal structural design, a result from the design ship structure activity

F.1.154

primary design in special areas

the fundamental and principal design of the ship not covered by the longitudinal and vertical structure

F.1.155

primary design of similar ships

the fundamental structural data of ships in the same category

F.1.156

primary structure

the first result during the design of the classification items with design approval preview

F.1.157

Produce and Approve Reference Documents* (A35)

the technical documentation for the ship, produced using production information

NOTE The output includes the loading and stability manual which is approved by the classification society.

F.1.158

Produce and Inspect a Ship (A3)

high-level activities such as produce, monitor and inspect ship production.

NOTE Inspect, means the controlling of all activities throughout the whole production life cycle of a ship.

F.1.159**Produce and Maintain Reference Documents* (A35)**

the technical documentation for the ship, produced using production information

NOTE The output includes the loading and stability manual.

F.1.160**Produce Modular Build Units (A32)**

producing the modular units which will make up the completed ship

NOTE They are produced from the steel-subsections and their production is controlled by the schedule, contract, the approved design, and any manufacturing restrictions. The results of the activity are the modular units which are assembled into the ship.

F.1.161**Produce Steel Sub-sections (A31)**

producing of the steel sub-sections which make up the structure of the completed ship.

NOTE This is controlled by the schedule, contract, the approved design, and any manufacturing restrictions.

F.1.162**product component information**

the technical data about the components that will be incorporated into the ship.

NOTE These are taken into consideration when the preliminary designs are being made.

F.1.163**production and delivery schedule***

the schedule according to which the ship is manufactured and delivered

F.1.164**production information**

information describing a product, e.g., dimensions, mechanical properties, workshop information

F.1.165**production planner**

the person making and carrying out production plans

F.1.166**production schedule***

data that controls the time from the production phase to delivery

F.1.167

propeller design*

the design of the propeller or propulsion as a result of the hydrodynamics and powering calculations

NOTE The design controls some of the machinery design activity.

F.1.168

quality assurance*

the rules applied by an organization within the shipyard that has the task to audit the shipyard organization and apply processes in a manner such that the quality of the resulting product is assured

F.1.169

Refine Thickness (A243424)

refining material strength of the plates

NOTE They are found and fixed as working information for the hull design.

F.1.170

Refine Working Information (A2443)

the most detailed and last stage of information to perform the production

F.1.171

Request a Ship* (A11)

the first activities of a ship owner when intending to order a ship

NOTE Having definite ideas regarding appearance and functionality of the vessel, the owner expresses these ideas in an inquiry to the shipyard.

F.1.172

Request Classification Items* (A2411)

collecting the approvable items from the classification society with information about the components and the request for changes

NOTE It produces a list of items which have to be calculated and approved.

F.1.173

request for production changes

changes that are requested to the ship design as a result of production experience or difficulties with the realization of the ship design

F.1.174

resistance and shaft power*

the result of the activity to estimate hydrodynamics and powering

NOTE Resistance and shaft power is a constraint on the creation of the preliminary hull form.

F.1.175**resources**

the organizations to give mechanism to all ship building activities

EXAMPLE 1 The shipyard, classification society, and engineering consultants.

F.1.176**schedule***

the plan for governing the timing of the production phases

NOTE Formed as a part of the final design process.

F.1.177**scrapping plan***

the document used to schedule the time and resources required to dismantle the ship

F.1.178**secondary structure design**

the structure detail design not defined in early design, as input for handling the plates and profiles

F.1.179**shell and longitudinal bulkheads walls**

the watertight skin of the ship and the bulkheads walls between transverse frames

F.1.180**ship**

a large waterborne vessel whose design, manufacture and lifecycle operation is governed by the principles of naval architecture and in accordance with international and classification society regulations

F.1.181**ship product model data**

the product data of the ship accumulated throughout its lifecycle

NOTE Because scrapping is part of the lifecycle, the ship is not an output, only the documented information and knowledge about the ship survives.

F.1.182**ship weight modifications**

modifications to ship weight due to the preliminary structure design

NOTE This is fed back to modify the preliminary hull form and revise the preliminary general arrangements.

F.1.183

shipyard

an organization that designs, builds, maintains, and repairs ships

F.1.184

Specify Ship (A1)

all activities associated with the production of a detailed specification of the ship prior to a contract being placed

F.1.185

Statutory and Class Survey* (A43)

preparative and executive tasks to perform a survey with focus on the statutory survey to meet the requirements of the class certificate

F.1.186

steel sub-sections

the sub-sections of the steel structure which are outfitted with the machinery and distribution systems before assembly

F.1.187

structural design

the design of the hull structure including hull, bulkheads, decks and stiffeners

F.1.188

structural design in special areas

the design of the hull structure not including longitudinal and vertical structure

F.1.189

survey report*

the report on the results of inspections of ship machinery, structure or other equipment

F.1.190

survey status*

the status of a survey at a particular point in time

F.1.191

technical documentation

the detailed information about material parts needed for producing the ship and system

NOTE In the case of maintenance the technical documentation of a system means part of the product description required to perform preventative maintenance, repair and failure analysis of that system.

F.1.192

technical requirements

the owner's specifications that must be realized by the completed ship

F.1.193**Test Ship (A34)**

activities which test the actual ship against the design, contract and rules and regulations

NOTE The structure is tested and sea trials are carried out. The test results are an output from this activity.

F.1.194**Test Structures (A341)**

activities which test steel structures against rules, regulations and the design

NOTE The output is the test result documentation.

F.1.195**Test Systems* (A342)**

activities which test the ship's systems including outfitting, machinery and mission systems against rules, regulations and the design

NOTE The output is the test result documentation.

F.1.196**Transform Piping and Electrical Design into Production Design* (A24422)**

a function to transform the conceptual data from design (CA) into material demand and assembling information

F.1.197**transportation need**

a constraint which determines the specification for the ship construction

F.1.198**updated survey status***

the updated status of a survey after carrying a statutory and class survey

F.1.199**WCOG information**

the information about weights and centres of gravity

F.1.200**weights and centres of gravity**

weights and centres of gravity necessary for further calculations

F.1.201**Work out Plans for Outfitting and Equipment* (A24423)**

the function that contains information for the various workforces in scheduling, controlling and coordinating their activities in their specific fields

F.1.202

working information

the information for production stage of the ship structures

F.2 Application activity model diagrams

The application activity model diagrams are given in 26. The graphical form of the application activity model is presented in the IDEF0 activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

figures F.2 through F.24 IDEF0

Figure F.1 describes the basic notation used in IDEF0 modelling. Each activity may be decomposed to provide more detail. If an activity has been decomposed, a separate diagram is included.

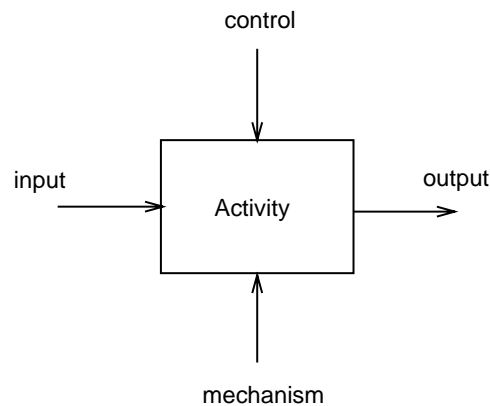


Figure F.1 – IDEF0 basic notation

As with any IDEF0 model, the AAM is dependent on a particular viewpoint and purpose. The purpose of the AAM is to describe the exchange of process engineering information.

Figure F.25 and figure F.26 indicate the node hierarchy of the AAM.

USED AT:	AUTHOR: Yuanxie Janke-Zhao PROJECT: AP218: Ship Structures NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 11/19/96 REV: 3.0	x	WORKING	READER	DATE	CONTEXT: Top
				DRAFT			
				RECOMMENDED			
				PUBLICATION			
<div><div><div>laws, rules and regulations</div><div>transportation need</div><div>manufacturing restrictions</div></div><div><div>historical data from previous designs</div><div>knowledge and experience</div><div>resources (shipyard, classification society, engineering consultants)</div></div><div><div>Perform Ship Life Cycle</div><div>A0</div></div><div><div>ship product model data</div><div>feedback</div></div></div>							
NODE: A-0		TITLE:			NUMBER: 1		

1098
Figure F.2 – Node A-0: Ship Life Cycle Description

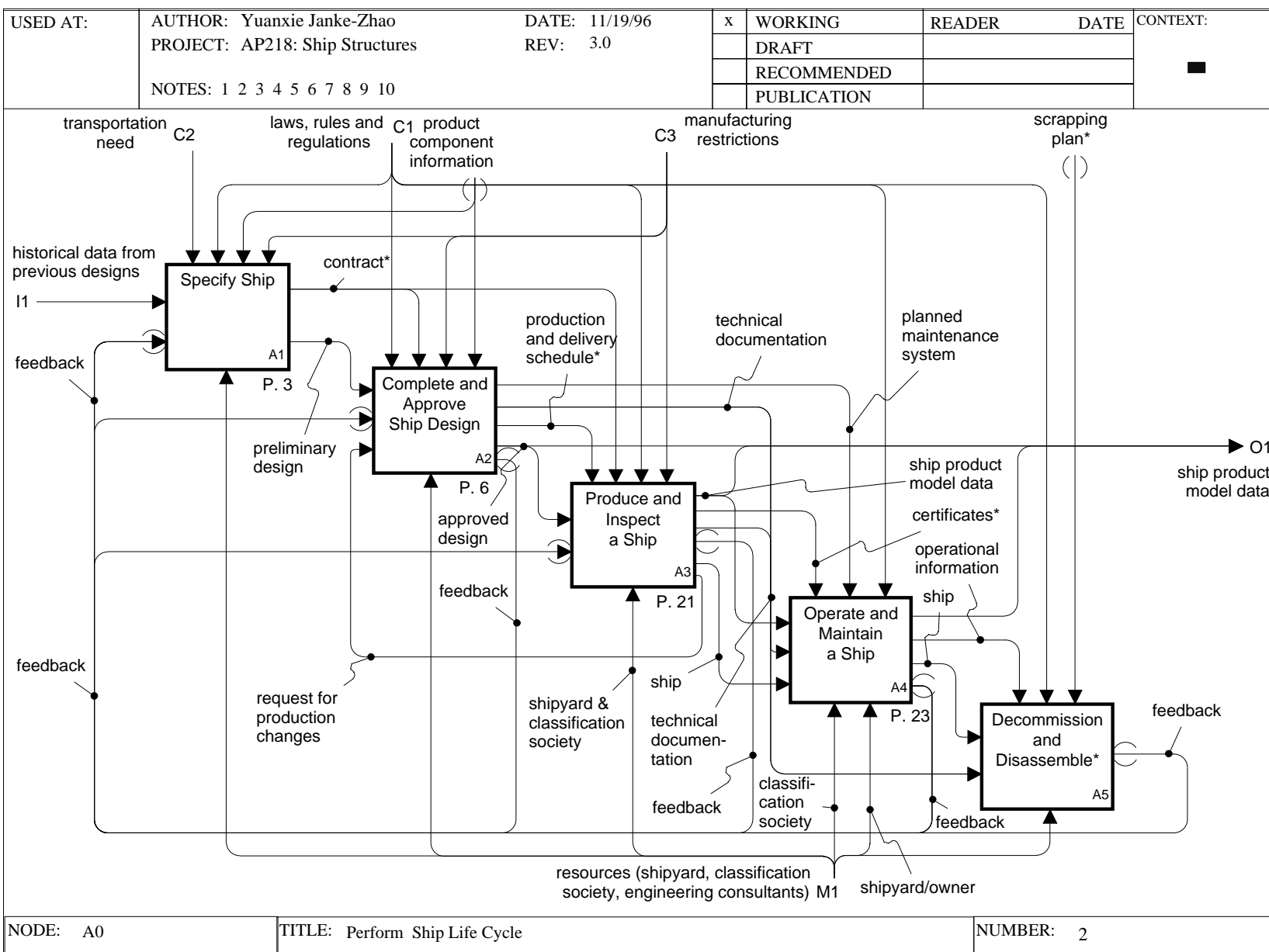


Figure F.3 – Node A0: Perform Ship Life Cycle

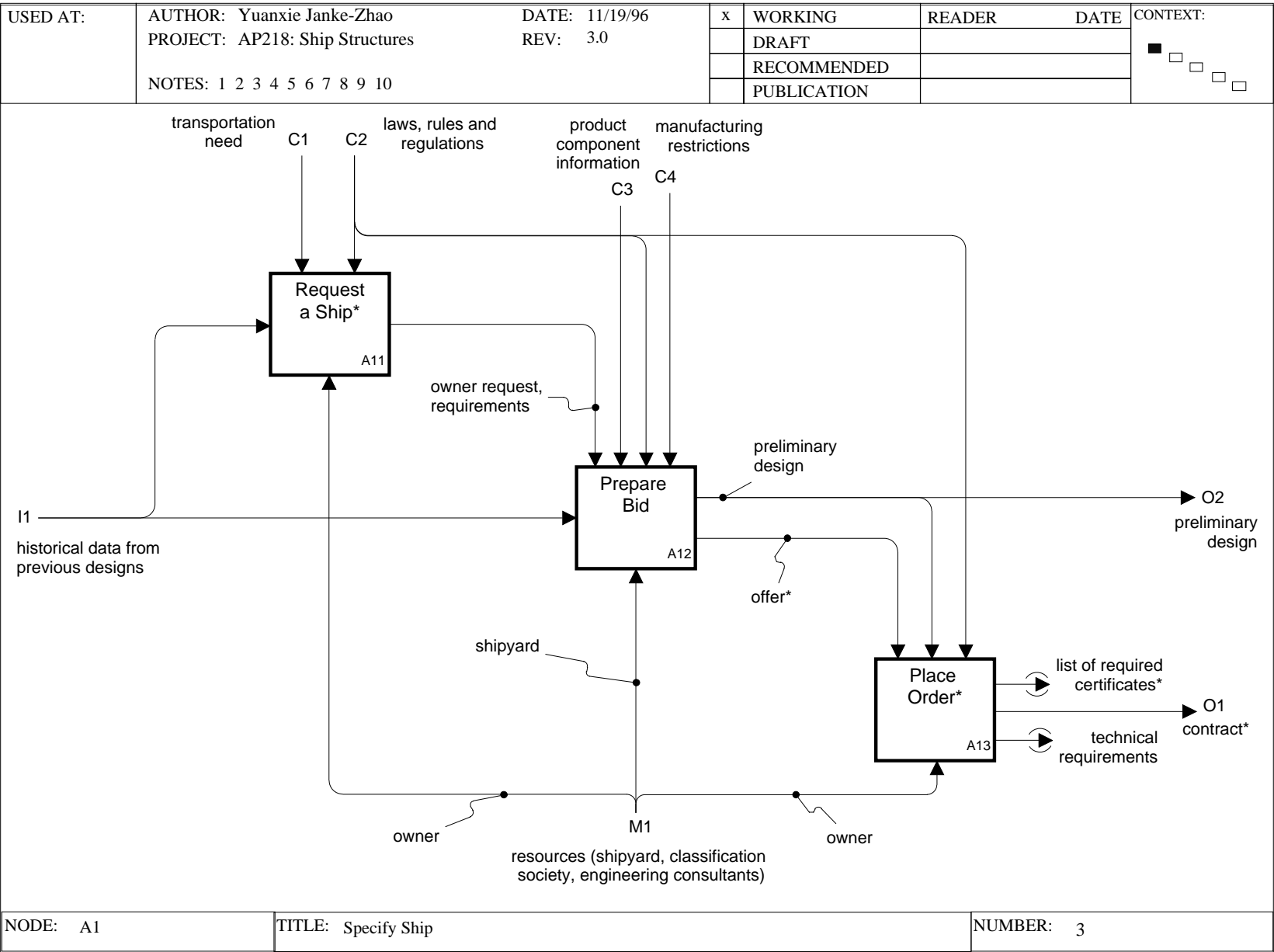


Figure F.4 – Node A1: Specify Ship

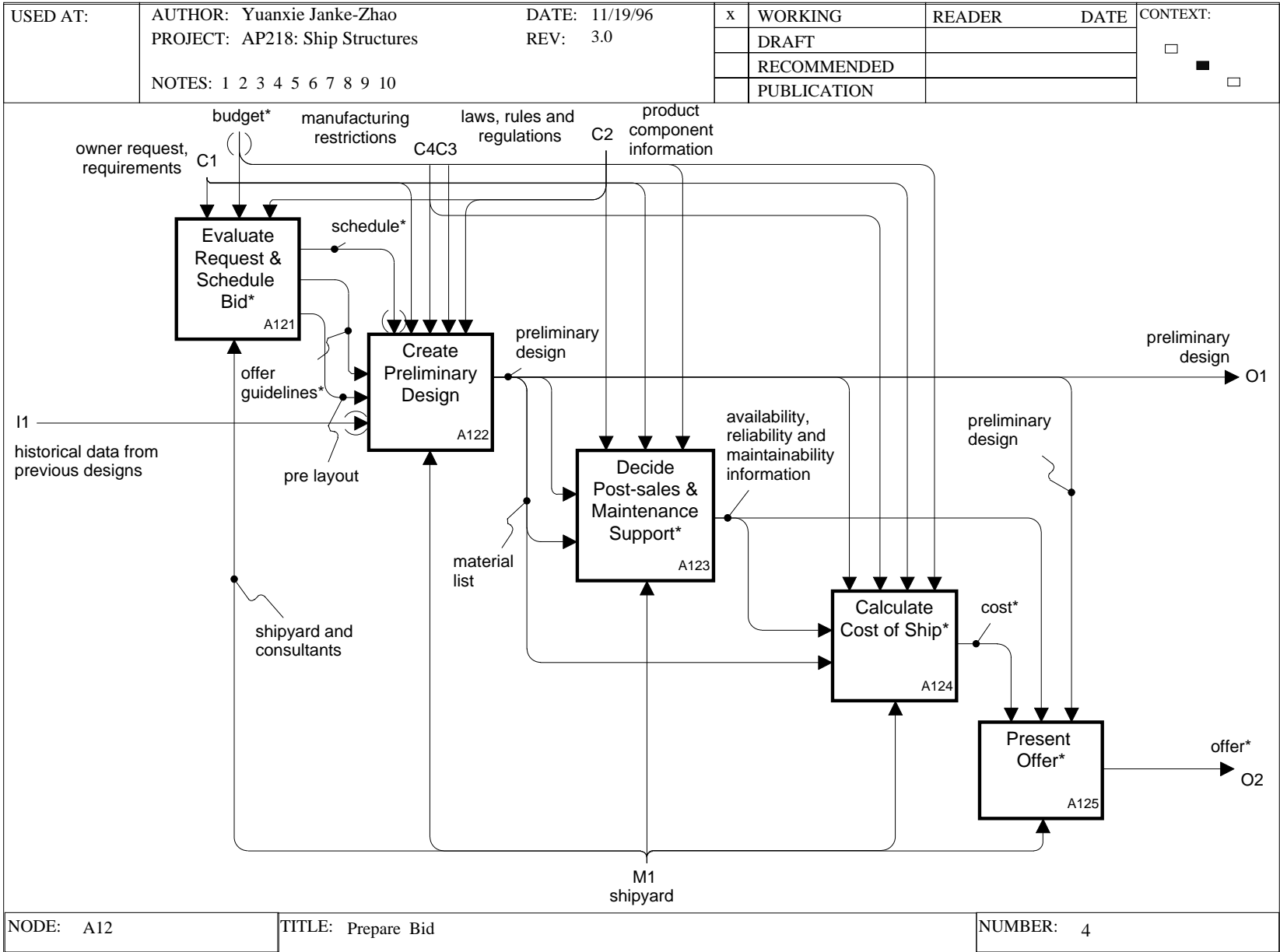
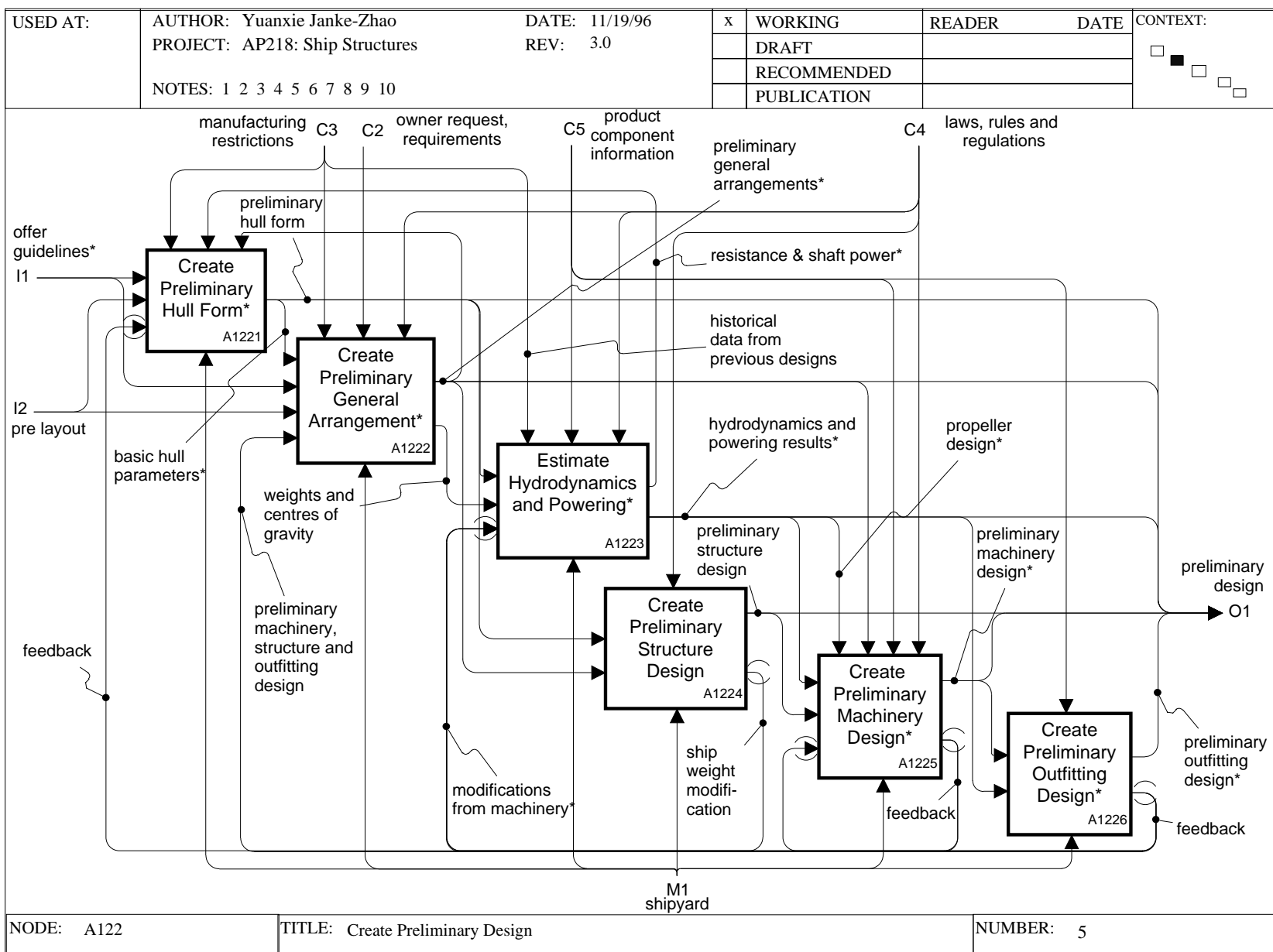


Figure F.5 – Node A12: Prepare Bid

Figure F.6 – Node A122: Create Preliminary Design



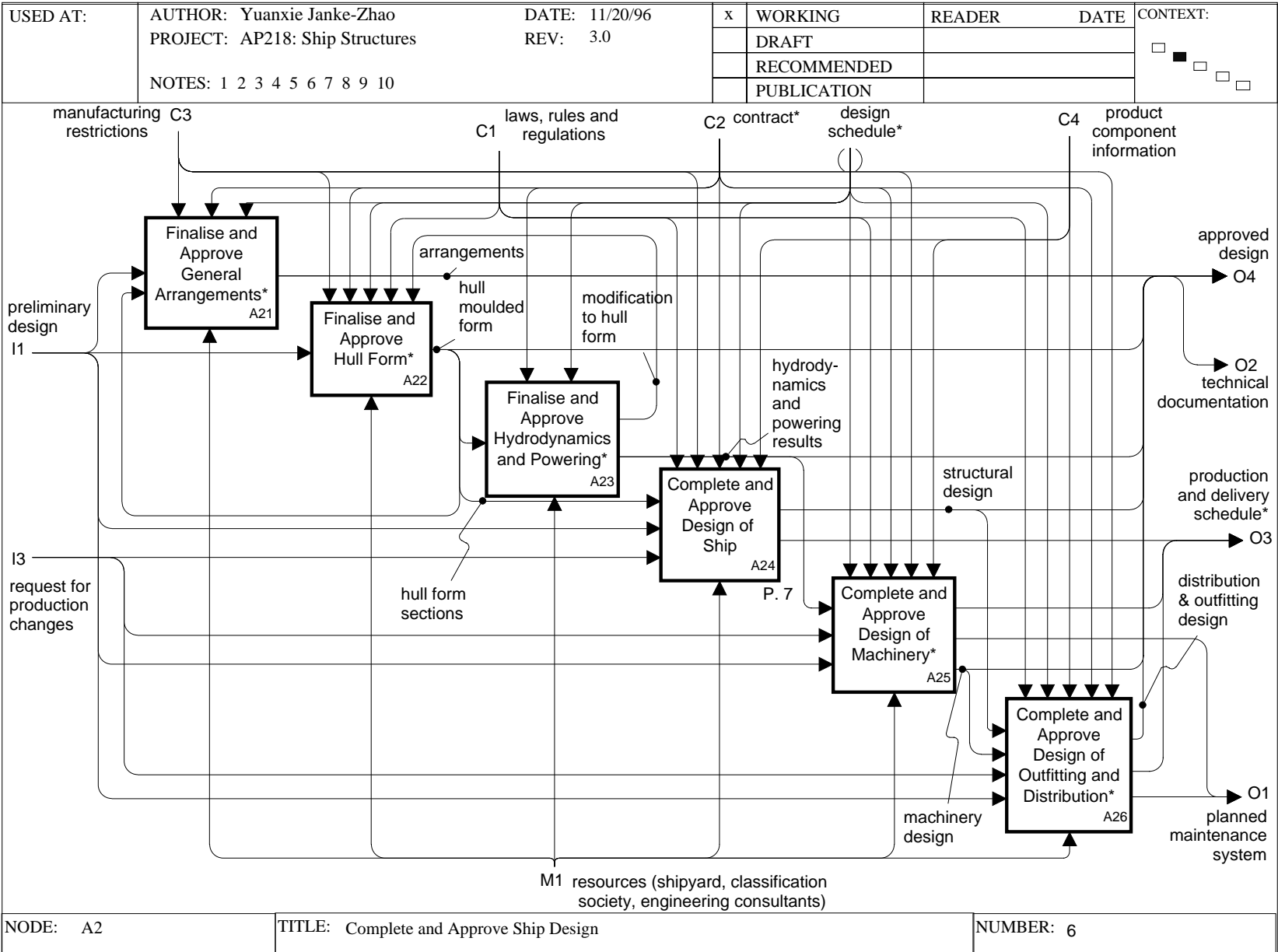
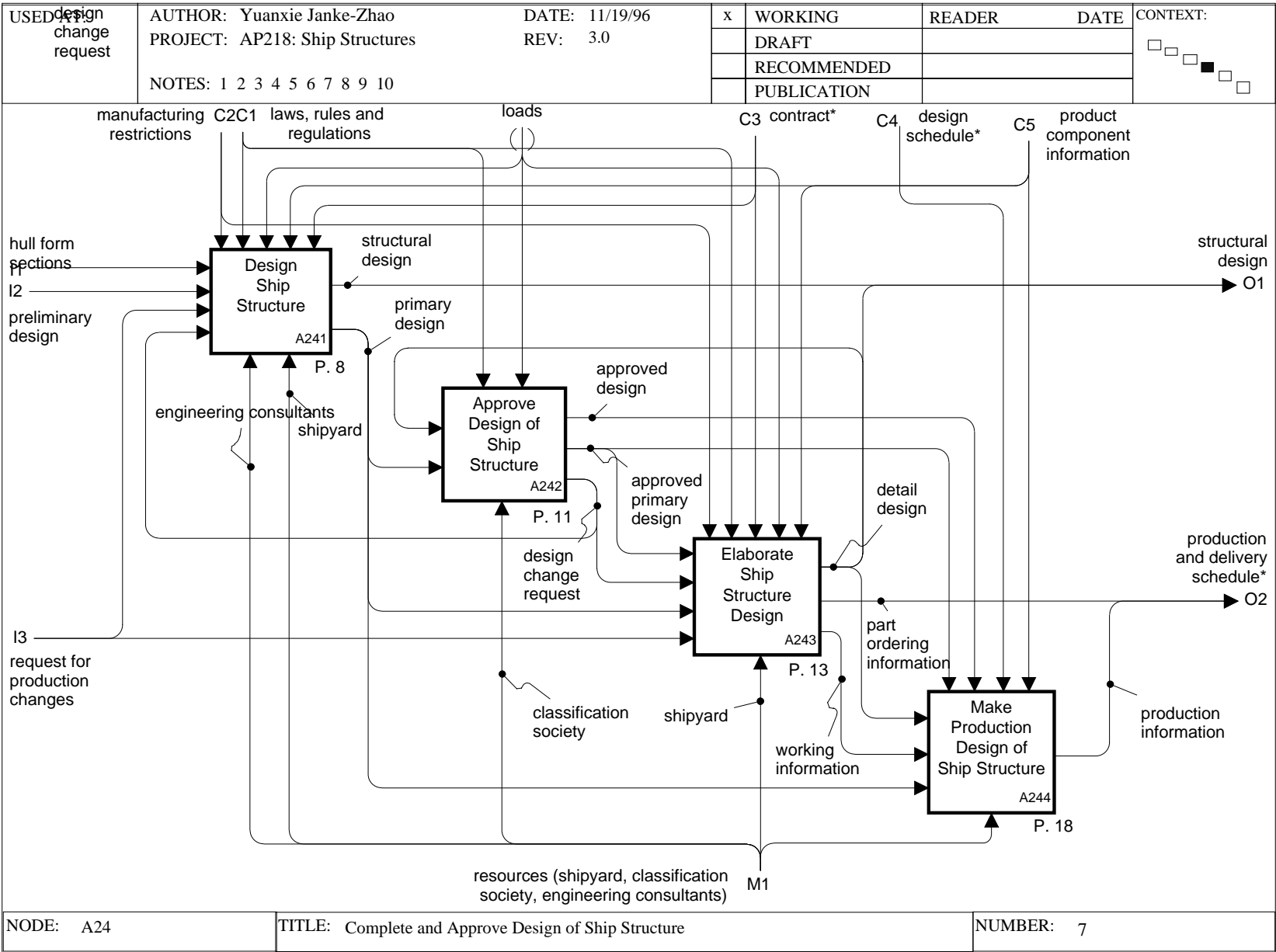


Figure F.7 – Node A2: Complete and Approve Ship Design
1103



1104
Figure F.8 – Node A24: Complete Design of Ship Structure

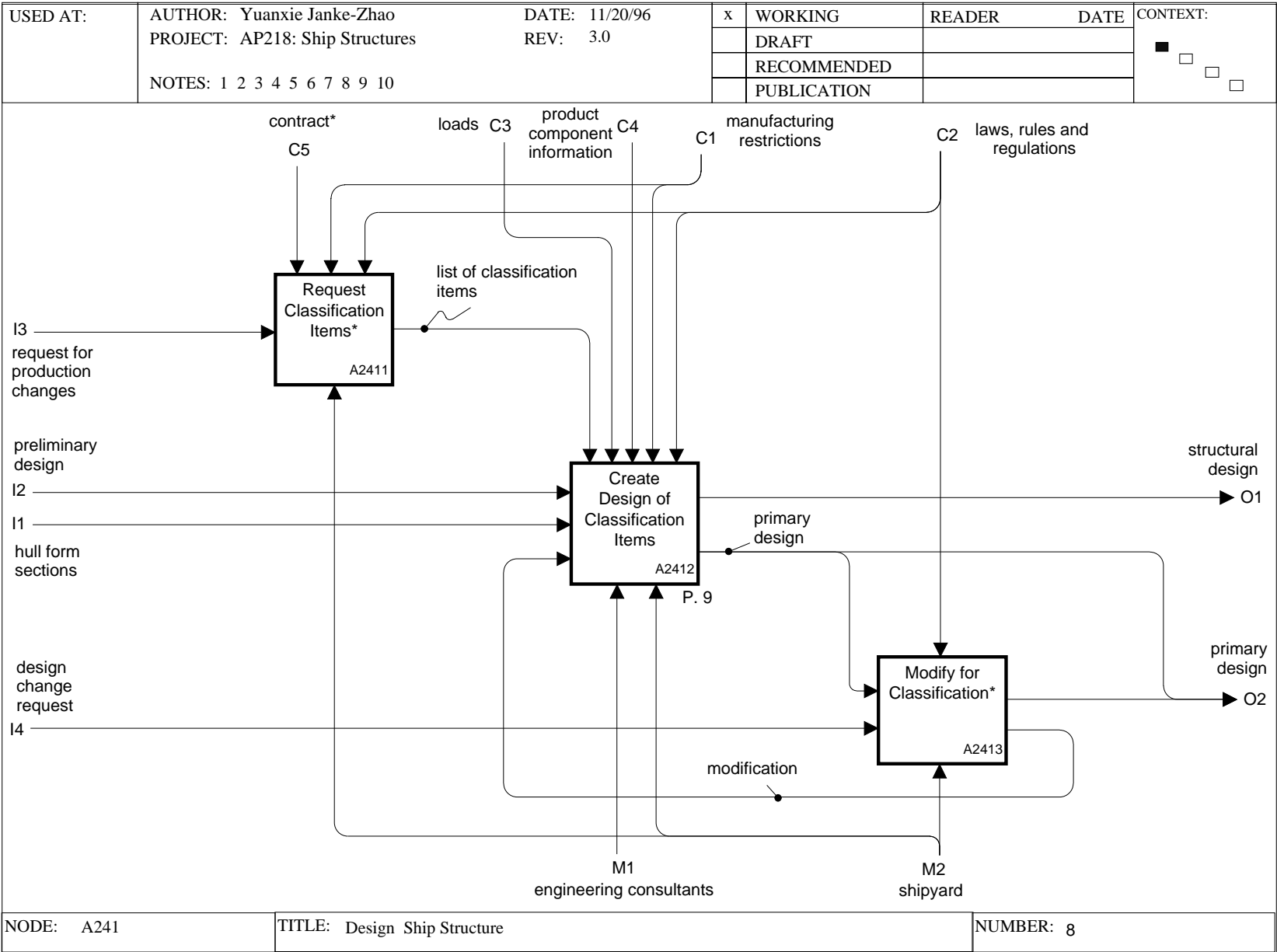
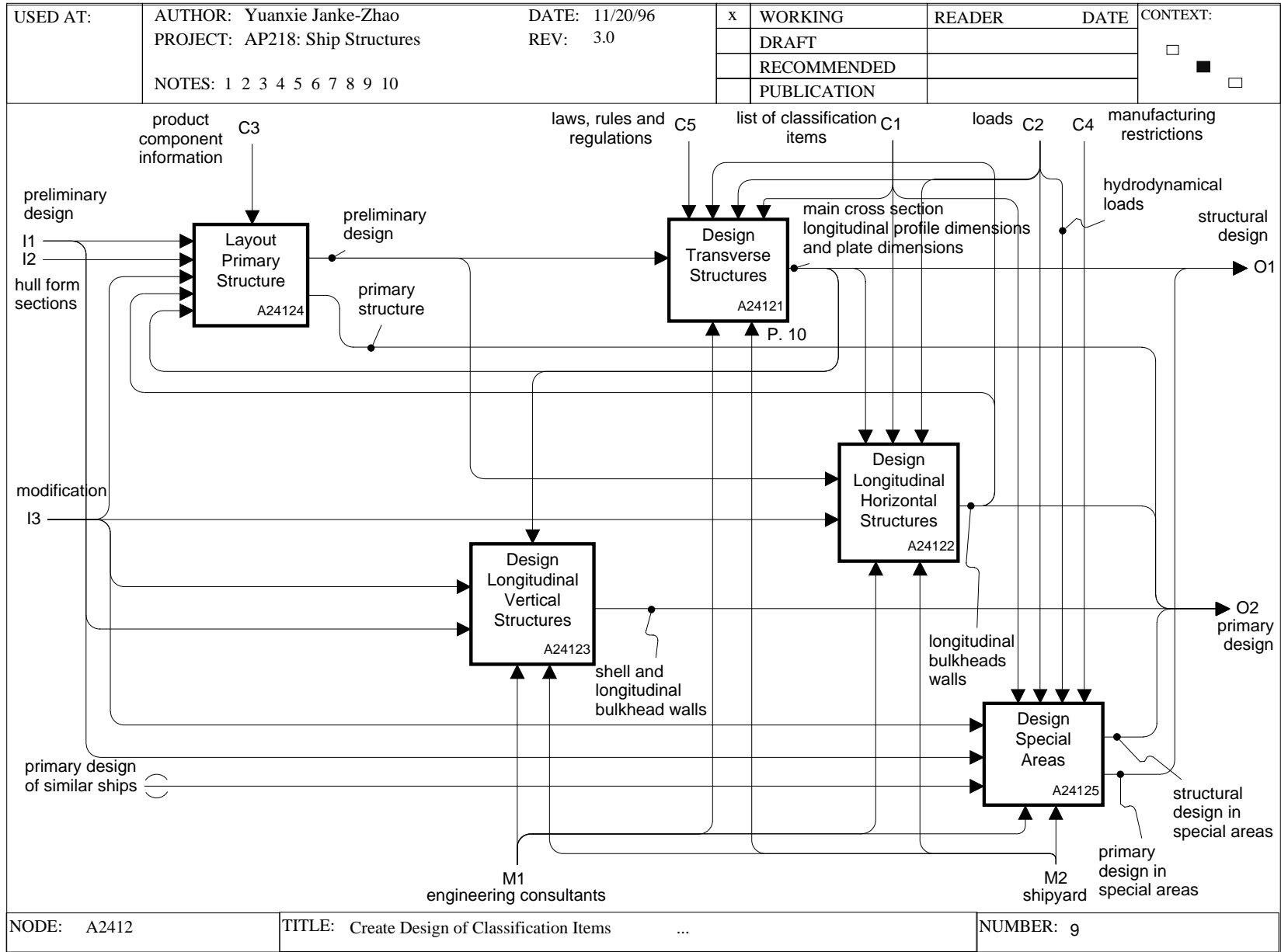


Figure F.9 – Node A241: Design Ship Structure



1106 Figure F.10 – Node A2412: Create Design of Classification Items

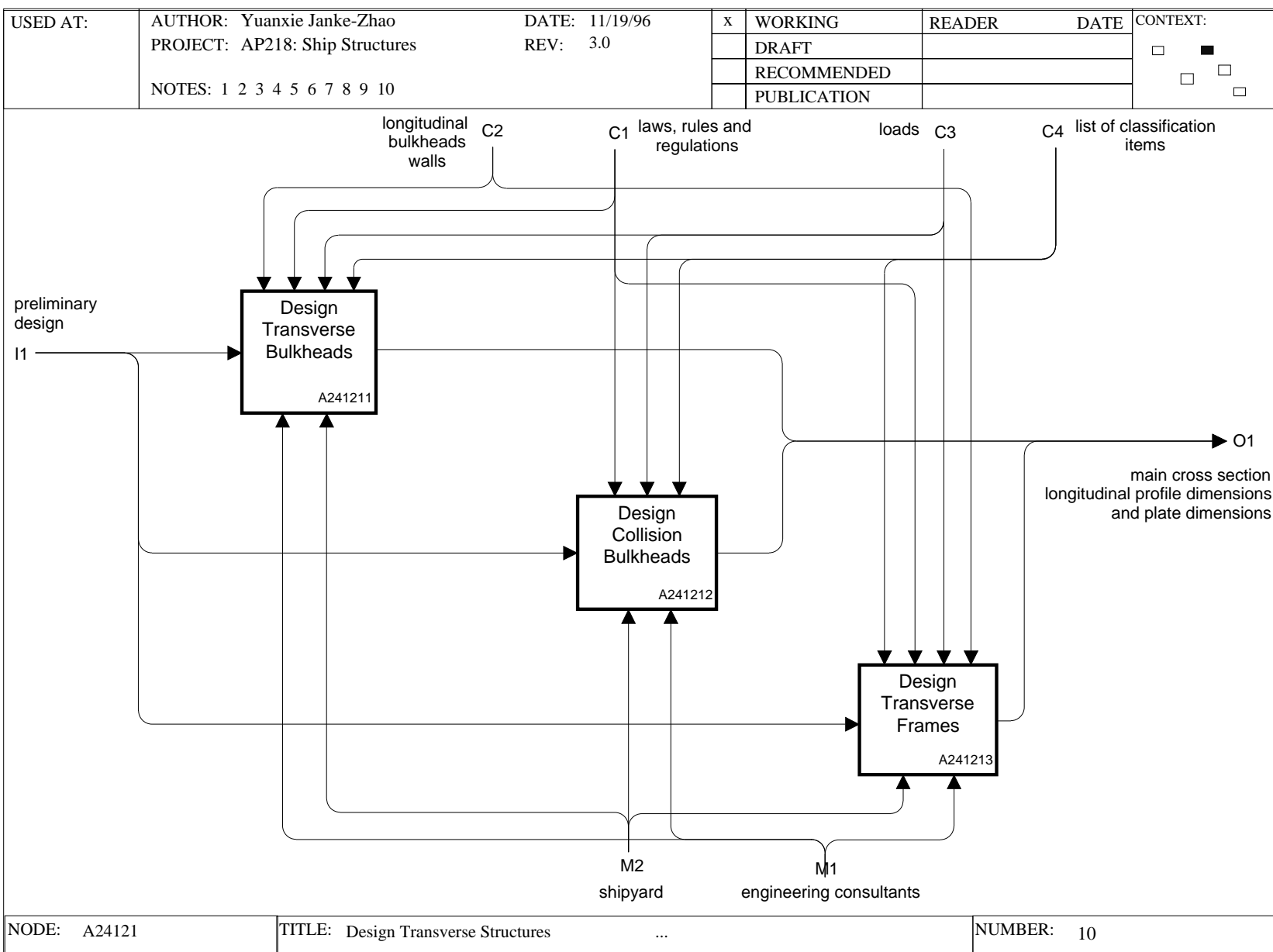
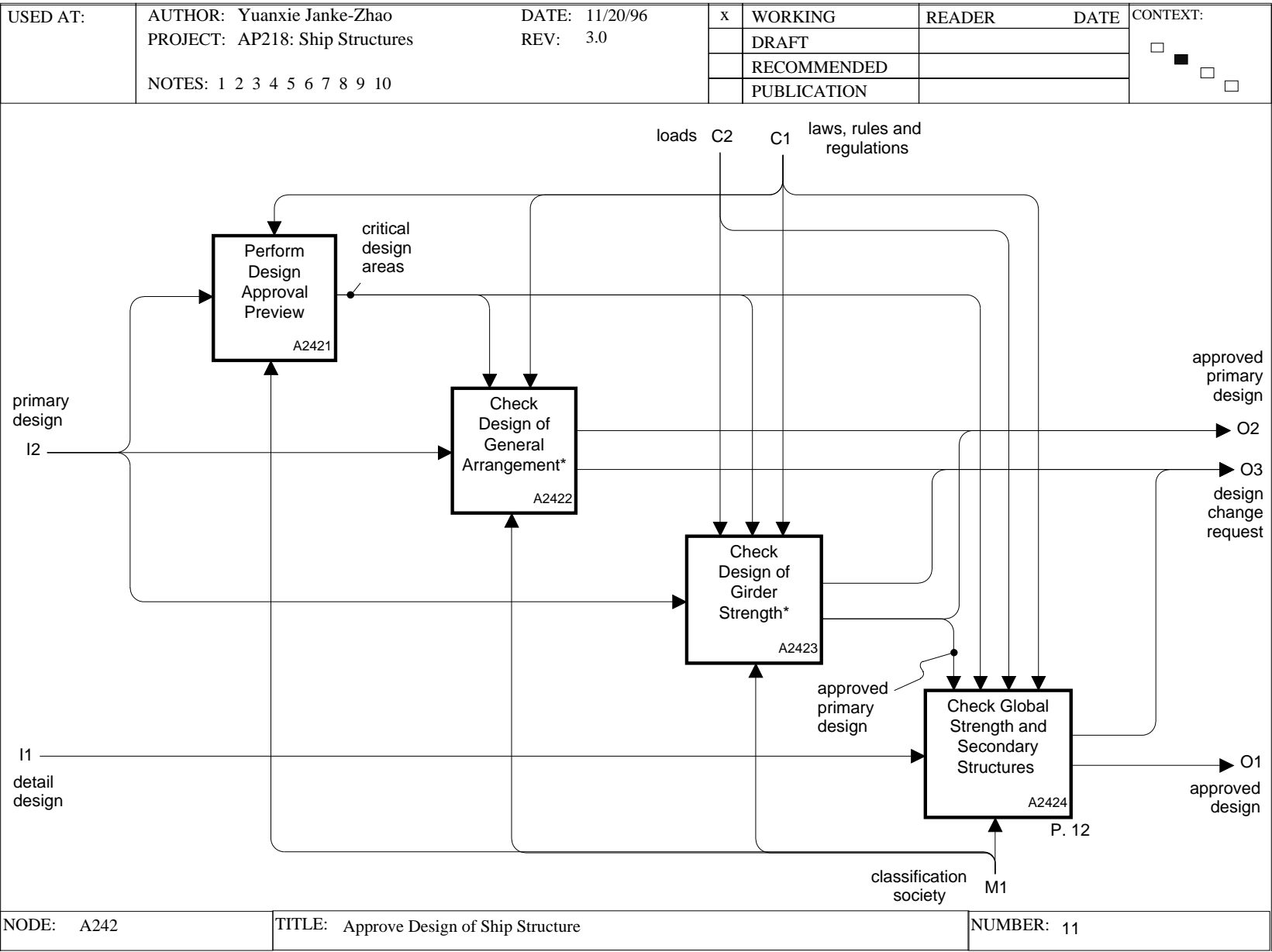


Figure F.11 – Node A24121: Design of Transverse Structure
1107

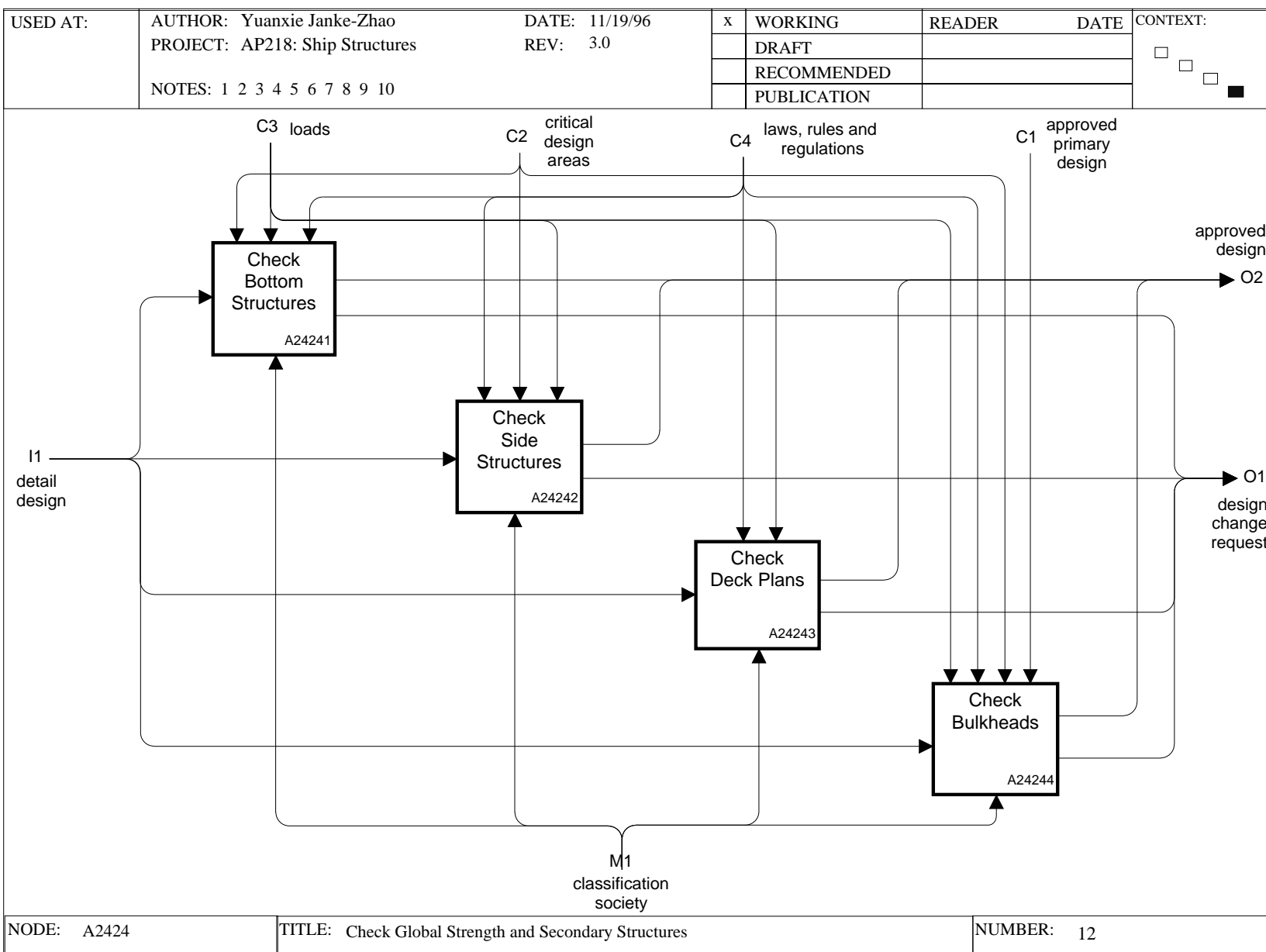


NODE: A242

TITLE: Approve Design of Ship Structure

NUMBER: 11

Figure F.12 – Node A242: Approve Design of Ship Structure

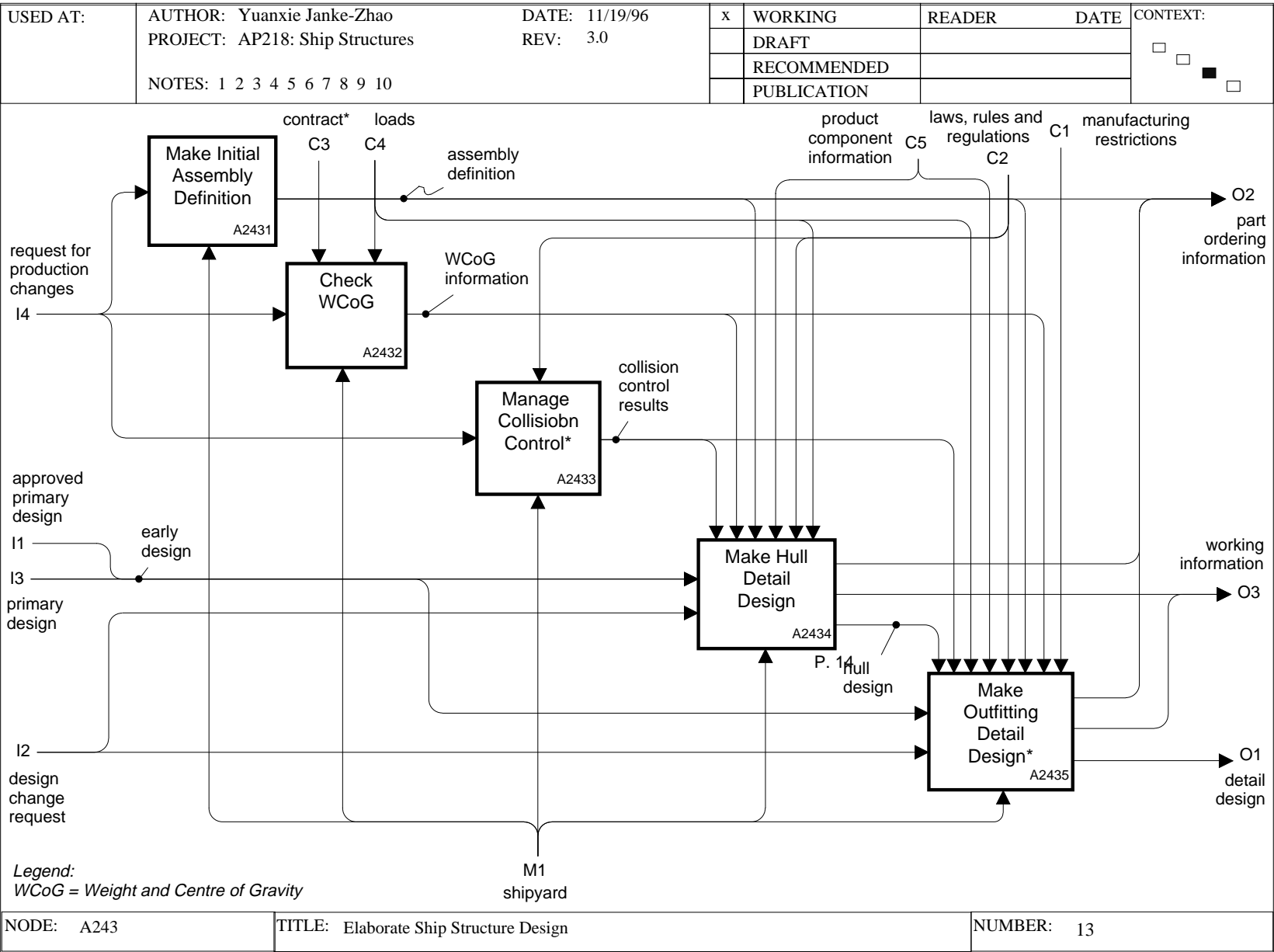


NODE: A2424

TITLE: Check Global Strength and Secondary Structures

NUMBER: 12

Figure F.13 – Node A2424: Check Global Strength and Secondary Structure 1109



1110 Figure F.14 – Node A243: Elaborate Ship Structure Design

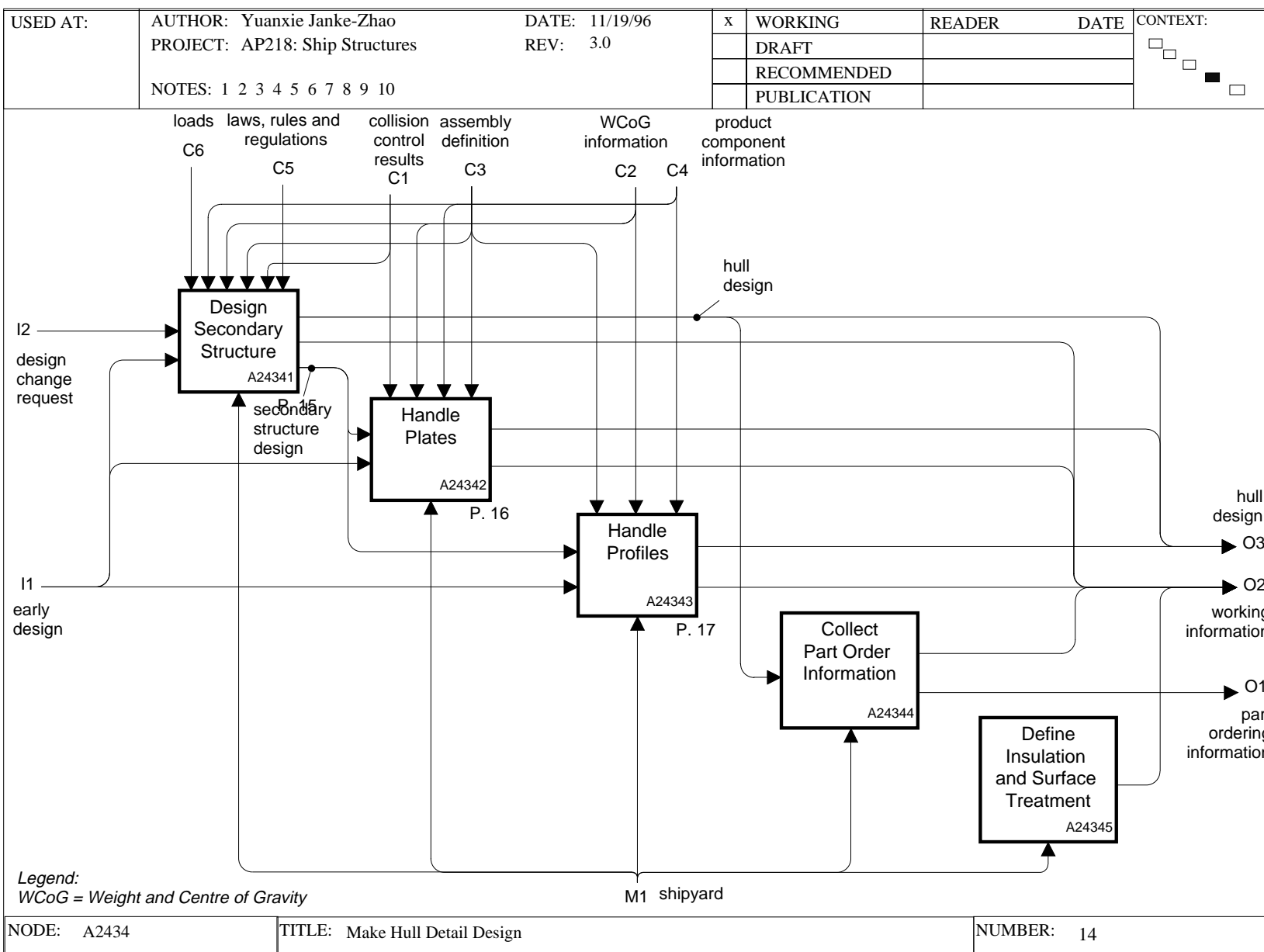
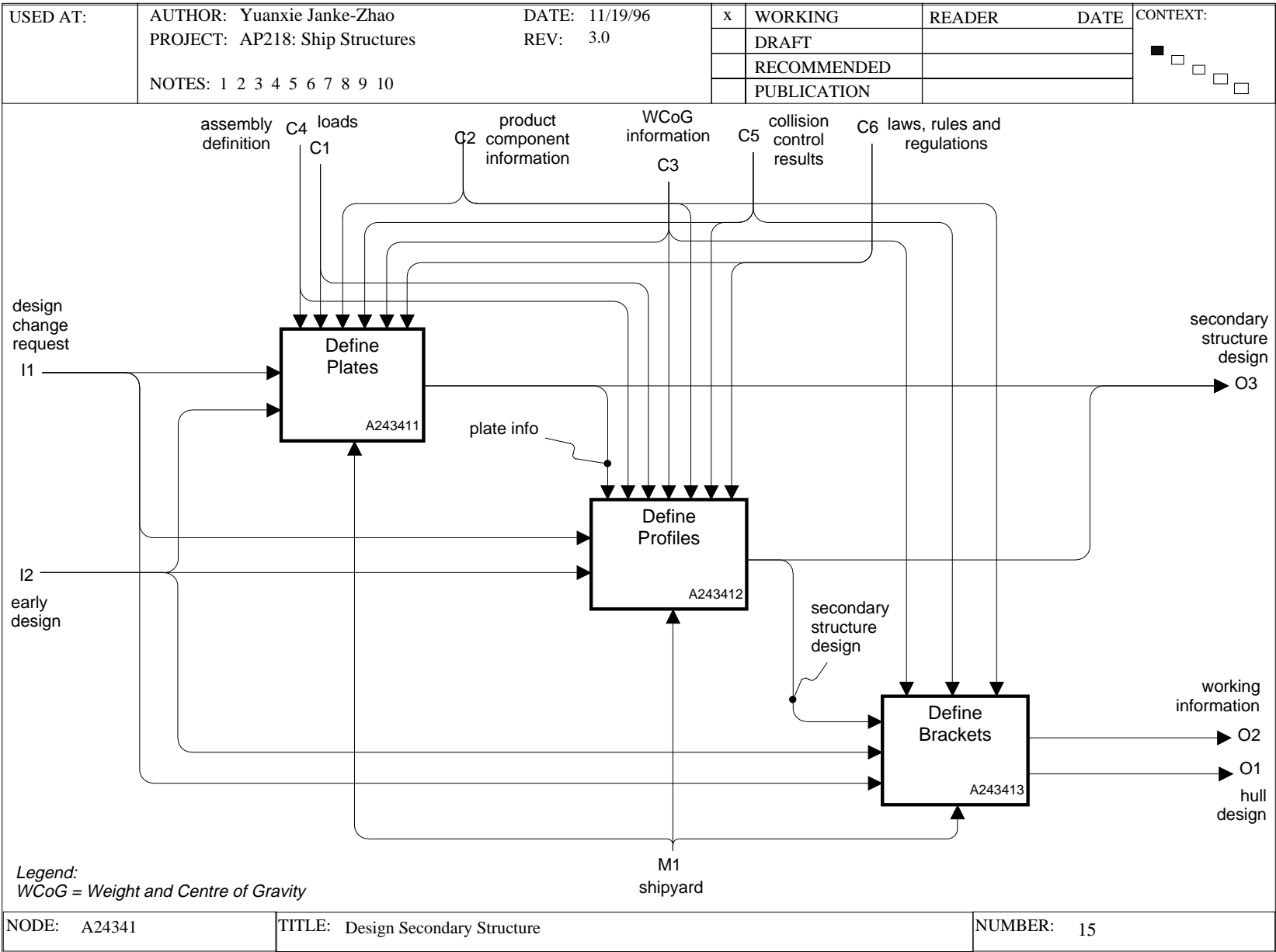


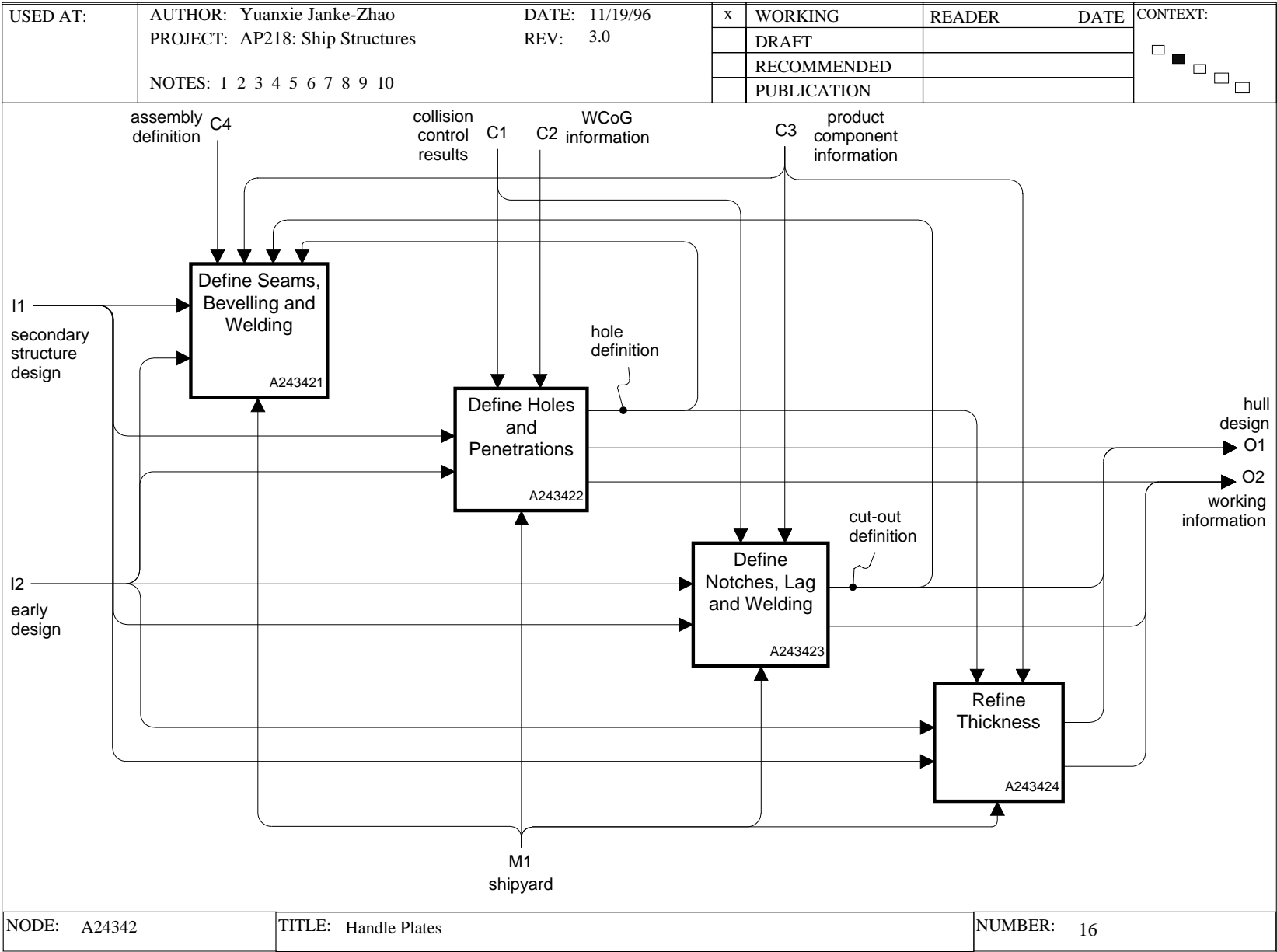
Figure F.15 – Node A2434: Make Hull Detail Design



1112 Figure F.16 – Node A24341: Design Secondary Structure

Figure F.17 – Node A24342: Handle Plates

1113



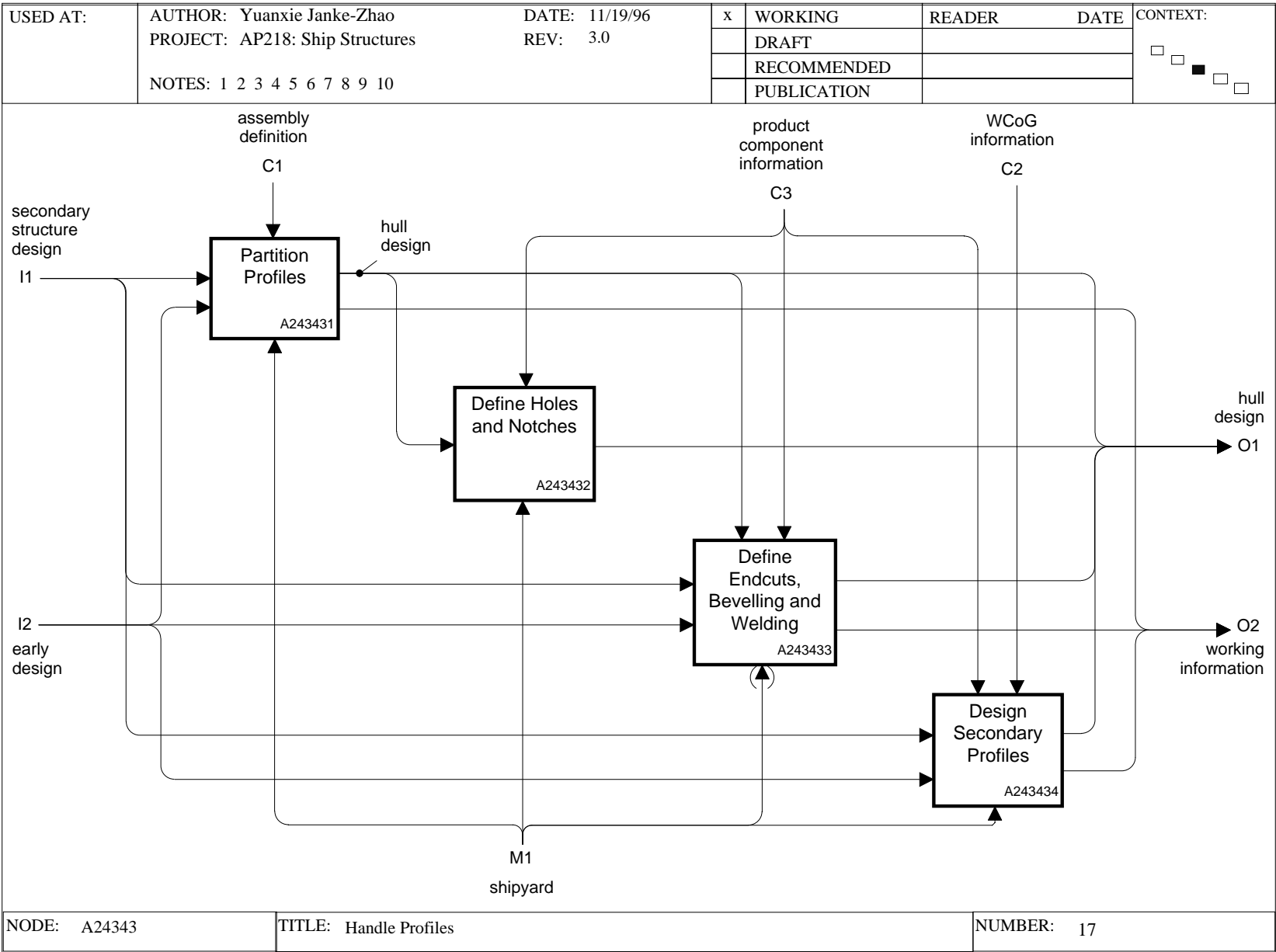


Figure F.18 – Node A24343: Handle Profiles

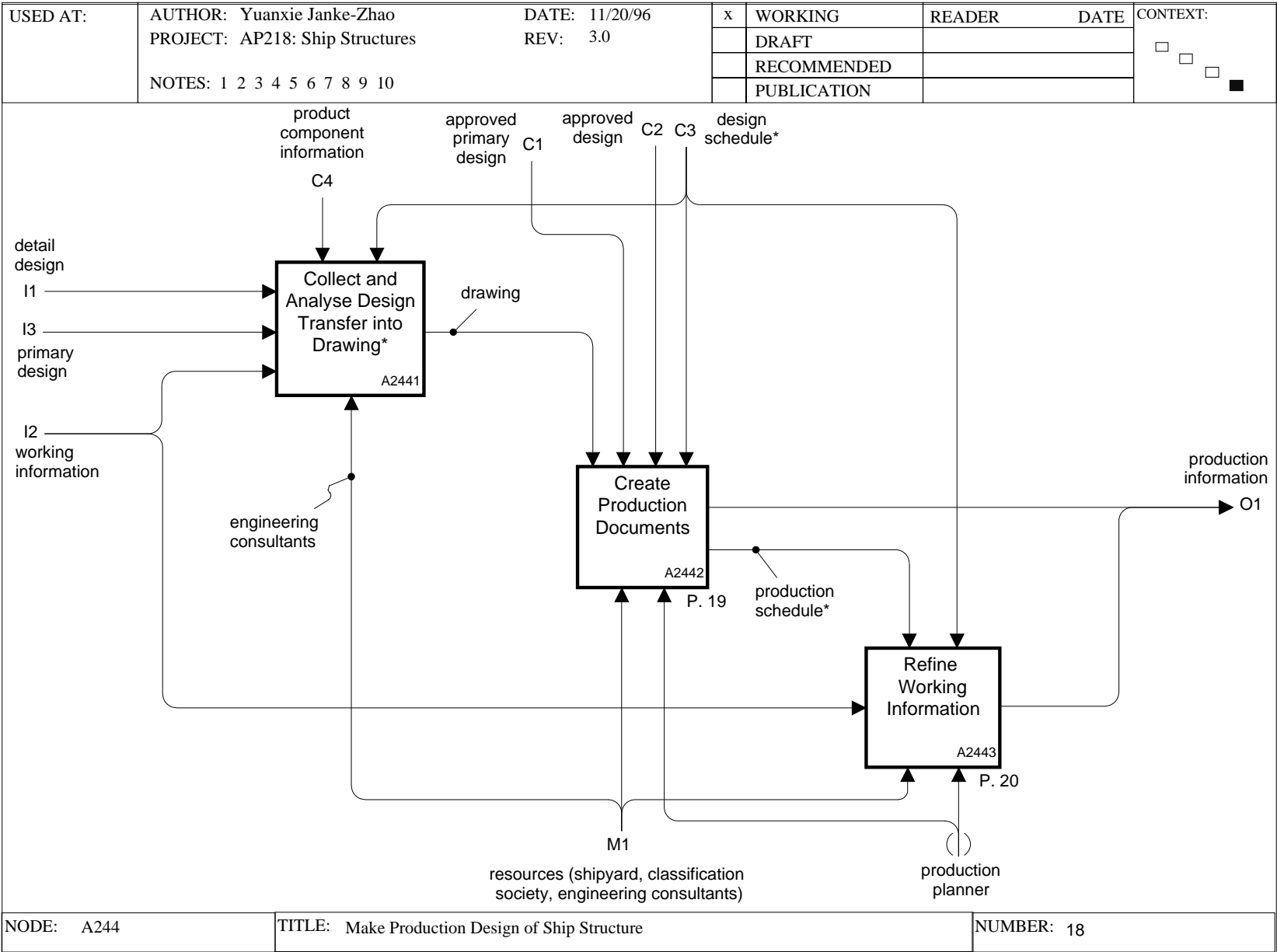
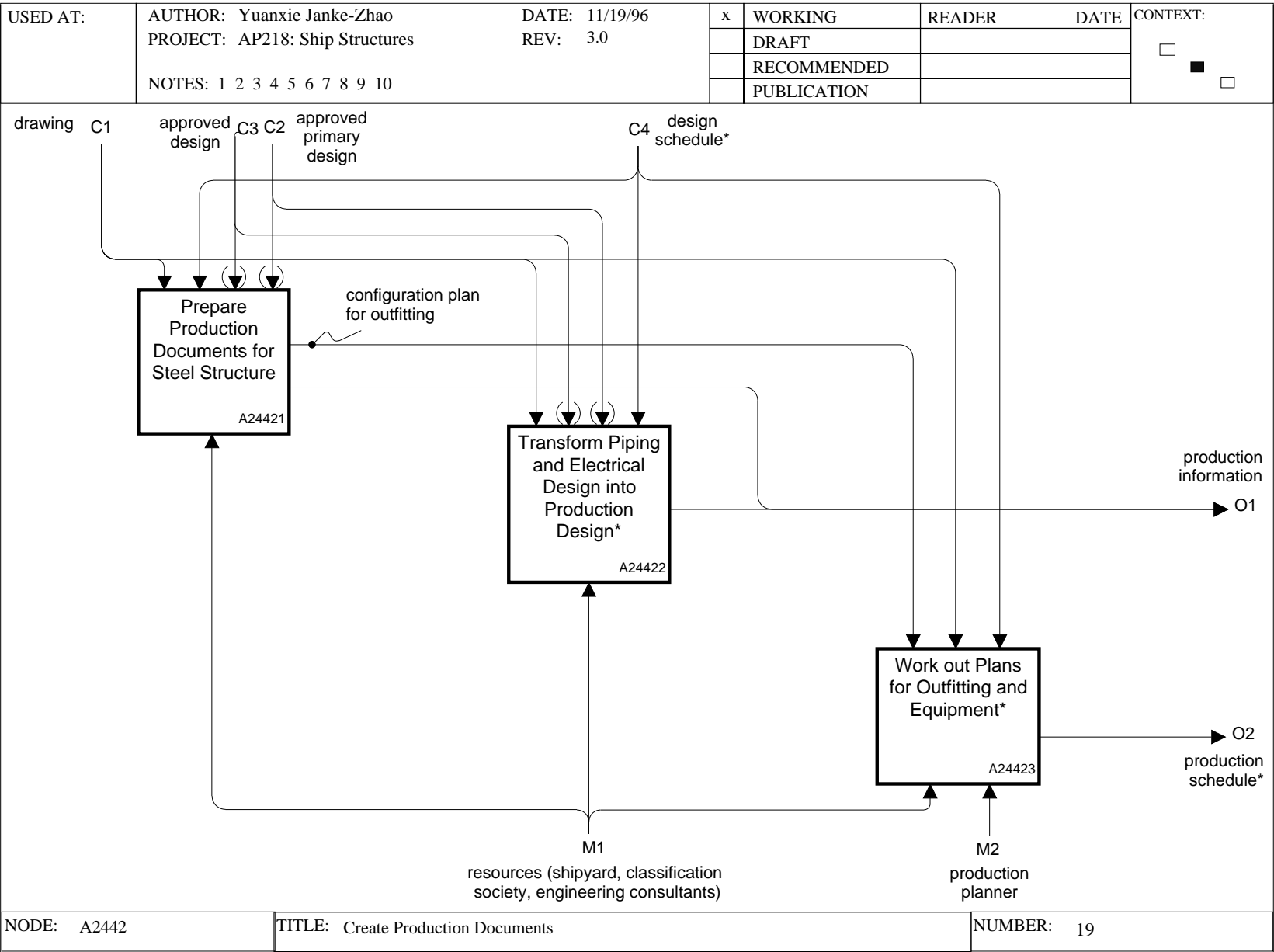


Figure F.19 – Node A244: Make Production Design of Ship Structure



1116 Figure F.20 – Node A2442: Create Production Documents

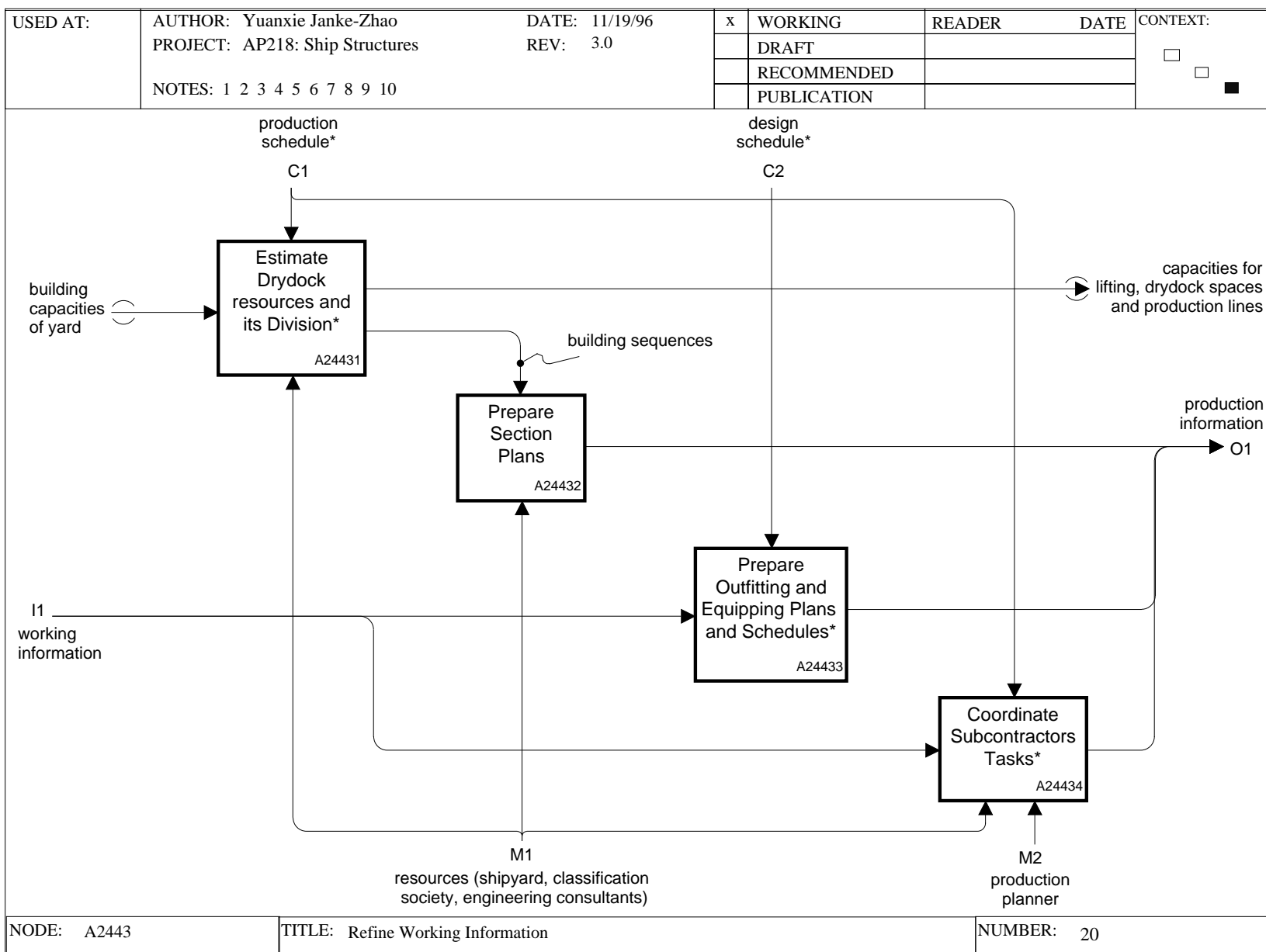


Figure F.21 – Node A2443: Refine Working Information 1117

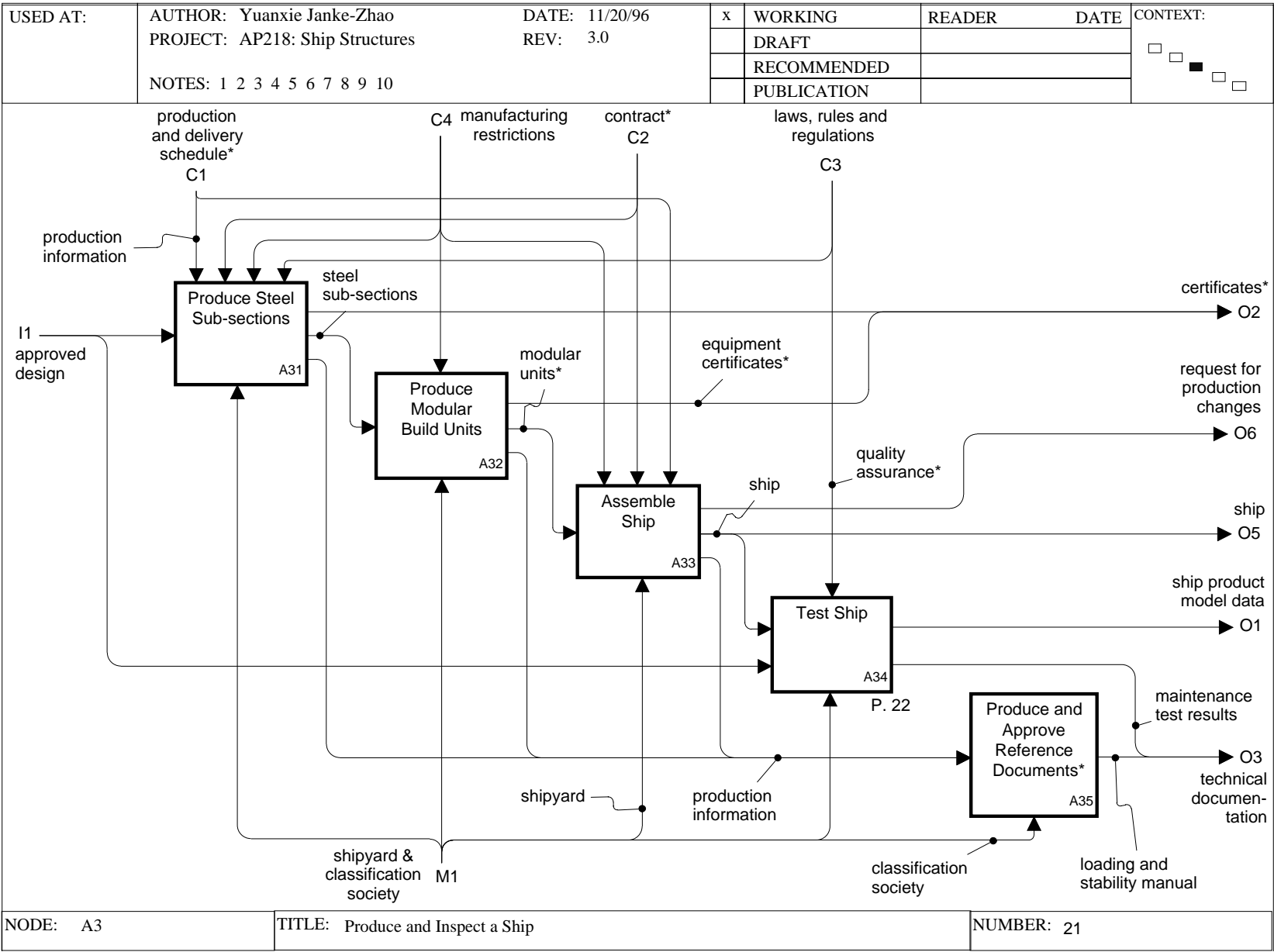


Figure F.22 – Node A3: Produce and Inspect a Ship

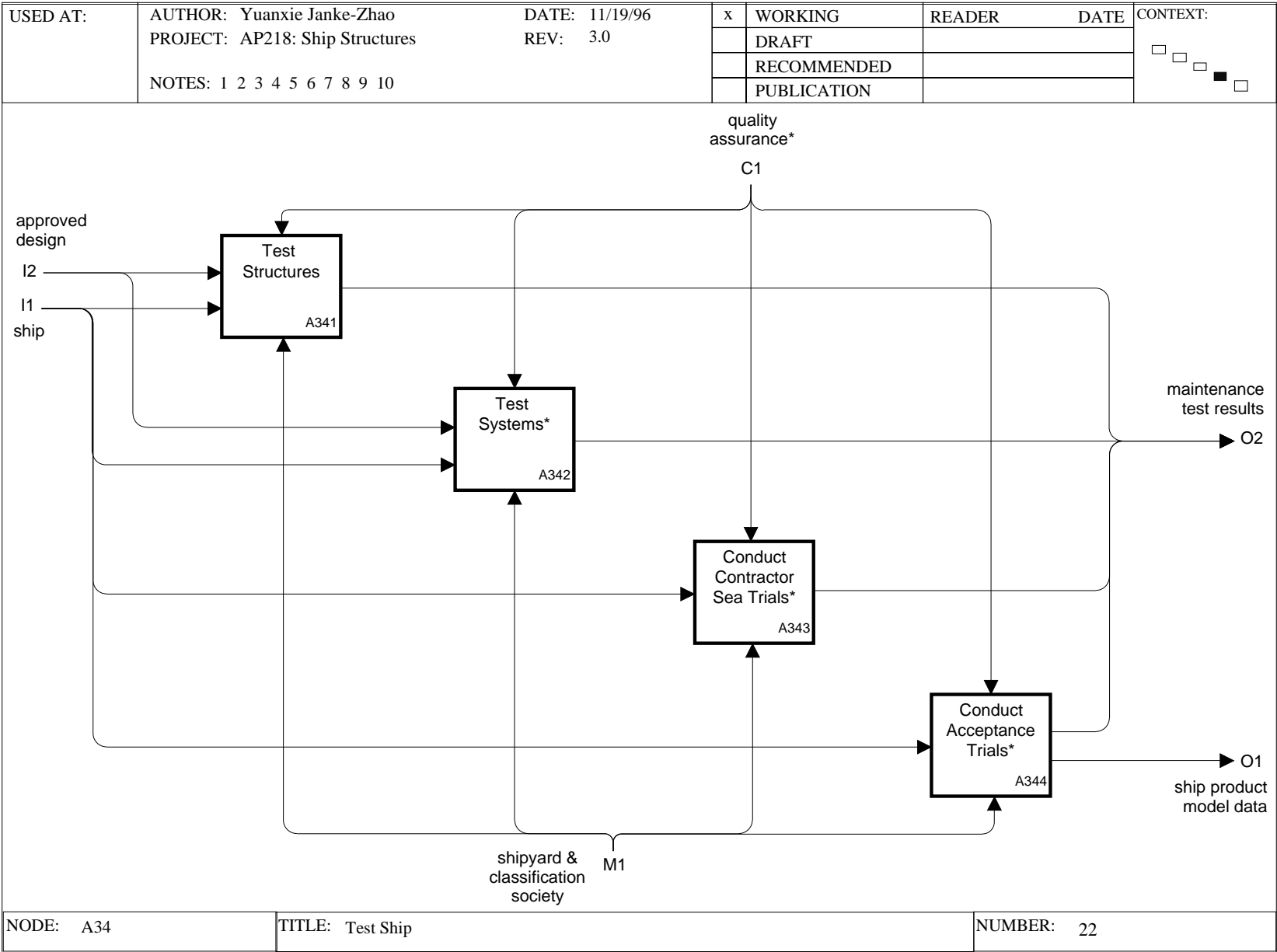


Figure F.23 – Node A34: Test a Ship

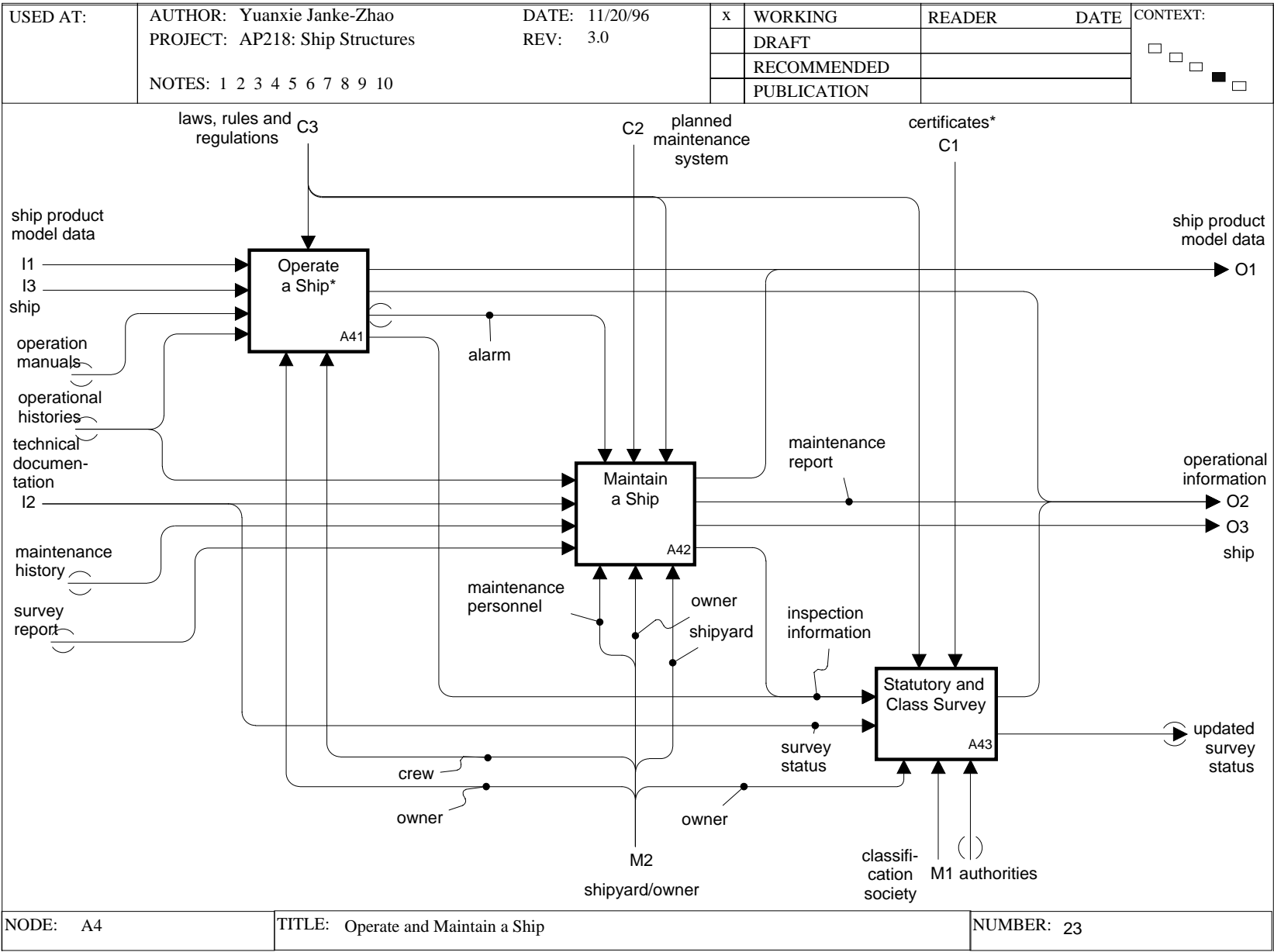


Figure F.24 – Node A4: Operate and Maintain a Ship

- [A0] Perform Ship Life Cycle
 - [A1] Specify Ship
 - [A11] Request a Ship*
 - [A12] Prepare Bid
 - [A121] Evaluate Request & Schedule Bid*
 - [A122] Create Preliminary Design
 - [A1221] Create Preliminary Hull Form*
 - [A1222] Create Preliminary General Arrangement*
 - [A1223] Estimate Hydrodynamics and Powering*
 - [A1224] Create Preliminary Structure Design
 - [A1225] Create Preliminary Machinery Design*
 - [A1226] Create Preliminary Outfitting Design*
 - [A123] Decide Post-sales & Maintenance Support*
 - [A124] Calculate Cost of Ship*
 - [A125] Present Offer*
 - [A13] Place Order*
 - [A2] Complete and Approve Ship Design
 - [A21] Finalise and Approve General Arrangements*
 - [A22] Finalise and Approve Hull Form*
 - [A23] Finalise and Approve Hydrodynamics and Powering*
 - [A24] Complete and Approve Design of Ship Structure
 - [A241] Design Ship Structure
 - [A2411] Request Classification Items*
 - [A2412] Create Design of Classification Items
 - [A24121] Design Transverse Structures
 - [A241211] Design Transverse Bulkheads
 - [A241212] Design Collision Bulkheads
 - [A241213] Design Transverse Frames
 - [A24122] Design Longitudinal Horizontal Structures
 - [A24123] Design Longitudinal Vertical Structures
 - [A24124] Layout Primary Structure
 - [A24125] Design Special Areas
 - [A2413] Modify for Classification*
 - [A242] Approve Design of Ship Structure
 - [A2421] Perform Design Approval Preview
 - [A2422] Check Design of General Arrangement*
 - [A2423] Check Design of Girder Strength*
 - [A2424] Check Global Strength and Secondary Structures
 - [A24241] Check Bottom Structures
 - [A24242] Check Side Structures
 - [A24243] Check Deck Plans
 - [A24244] Check Bulkheads
 - [A243] Elaborate Ship Structure Design
 - [A2431] Make Initial Assembly Definition
 - [A2432] Check WCoG
 - [A2433] Manage Collision Control*
 - [A2434] Make Hull Detail Design

Figure F.25 – Node Hierarchy of the Application Activity Model

- [A24341] Design Secondary Structure
 - [A243411] Define Plates
 - [A243412] Define Profiles
 - [A243413] Define Brackets
- [A24342] Handle Plates
 - [A243421] Define Seams, Bevelling and Welding
 - [A243422] Define Holes and Penetrations
 - [A243423] Define Notches, Lag and Welding
 - [A243424] Refine Thickness
- [A24343] Handle Profiles
 - [A243431] Partition Profiles
 - [A243432] Define Holes and Notches
 - [A243433] Define Endcuts, Bevelling and Welding
 - [A243434] Design Secondary Profiles
- [A24344] Collect Part Order Information
- [A24345] Define Insulation and Surface Treatment
- [A2435] Make Outfitting Detail Design*
- [A244] Make Production Design of Ship Structure
 - [A2441] Collect and Analyze Design Transfer into Drawings*
 - [A2442] Create Production Documents
 - [A24421] Prepare Production Documents for Steel Structure
 - [A24422] Transform Piping and Electrical Design into Production Design*
 - [A24423] Work out Plans for Outfitting and Equipment*
 - [A2443] Refine Working Information
 - [A24431] Estimate Drydock resources and its Division*
 - [A24432] Prepare Section Plans
 - [A24433] Prepare Outfitting and Equipping Plans and Schedules*
 - [A24434] Coordinate Subcontractors Tasks*
- [A25] Complete and Approve Design of Machinery*
- [A26] Complete and Approve Design of Outfitting and Distribution*
- [A3] Produce and Inspect a Ship
 - [A31] Produce Steel Sub-sections
 - [A32] Produce Modular Build Units
 - [A33] Assemble Ship
 - [A34] Test Ship
 - [A341] Test Structures
 - [A342] Test Systems*
 - [A343] Conduct Contractor Sea Trials*
 - [A344] Conduct Acceptance Trials*
 - [A35] Produce and Approve Reference Documents*
- [A4] Operate and Maintain a Ship
 - [A41] Operate a Ship*
 - [A42] Maintain a Ship
 - [A43] Statutory and Class Survey
- [A5] Decommission and Disassemble*

Figure F.26 – Node Hierarchy of the Application Activity Model (continued)

Annex G

(informative)

Application reference model

This annex provides the application reference model for this part of ISO 10303 and is given in figures G.1 through G.69. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4. The graphical form of the application reference model is presented in the EXPRESS-G format. The application reference model is independent from any implementation method.

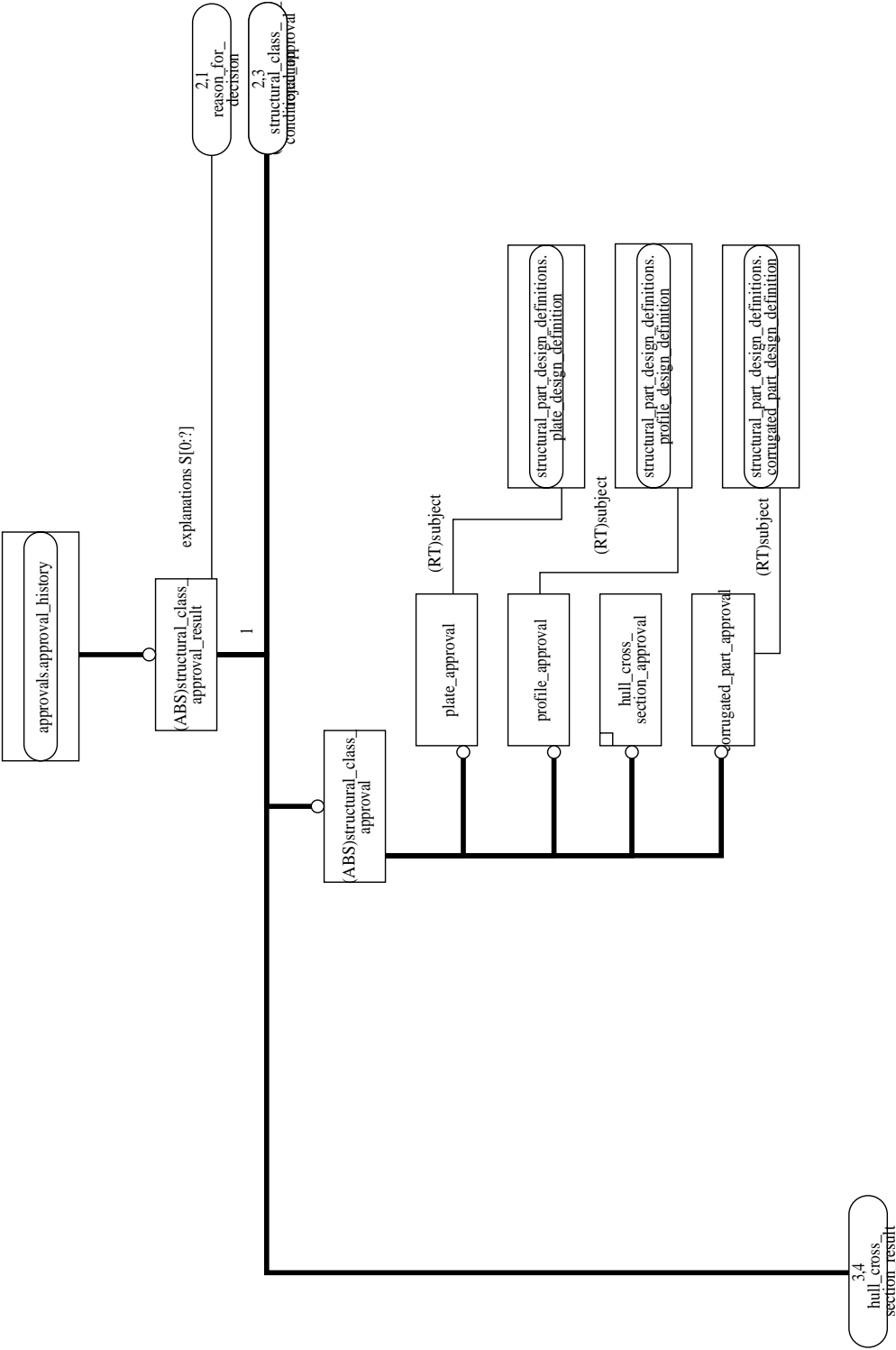


Figure G.1 – class_approvals_structure schema (1/3) in the class_approvals UoF

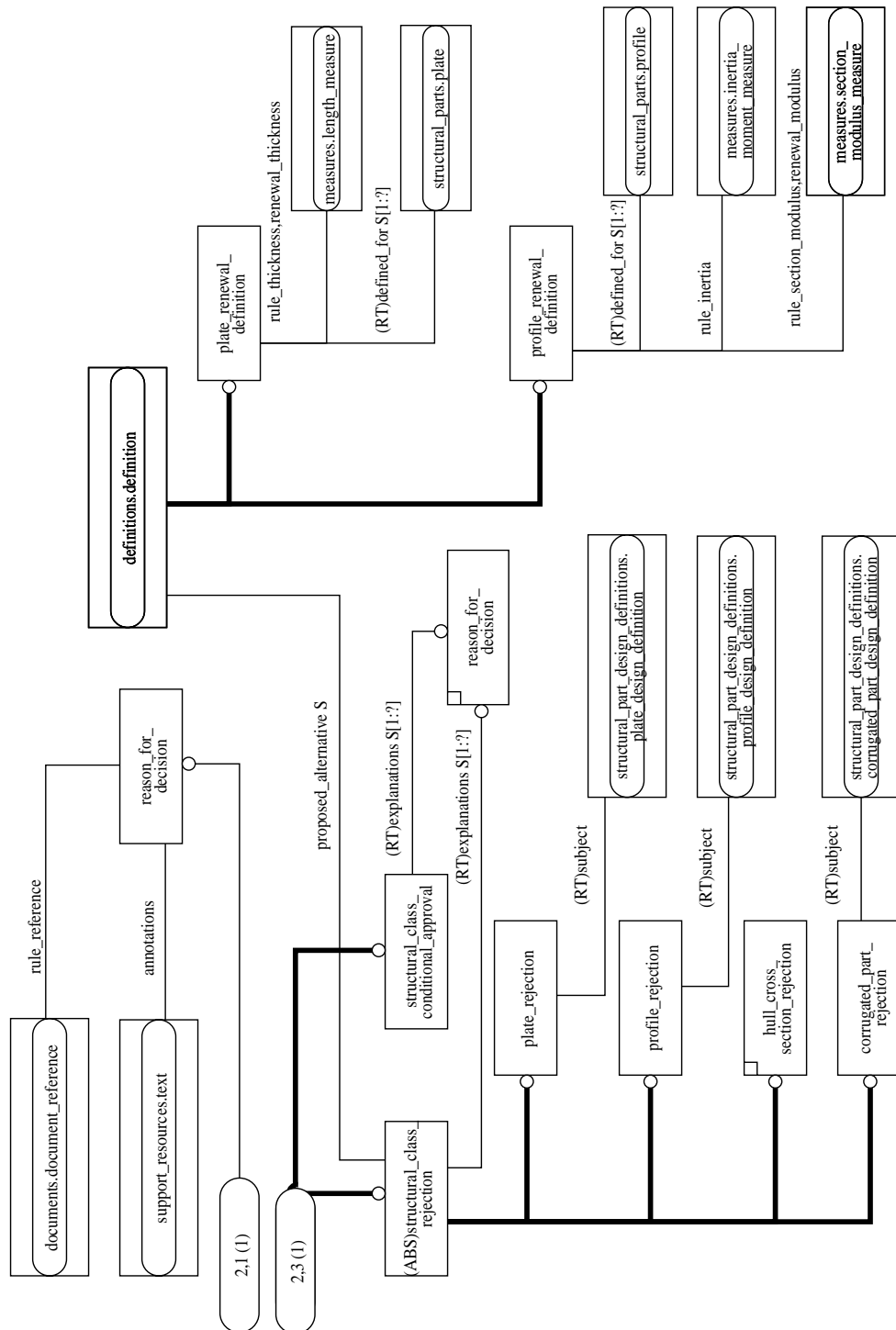


Figure G.2 – class_approvals_structure schema (2/3) in the class_approvals UoF

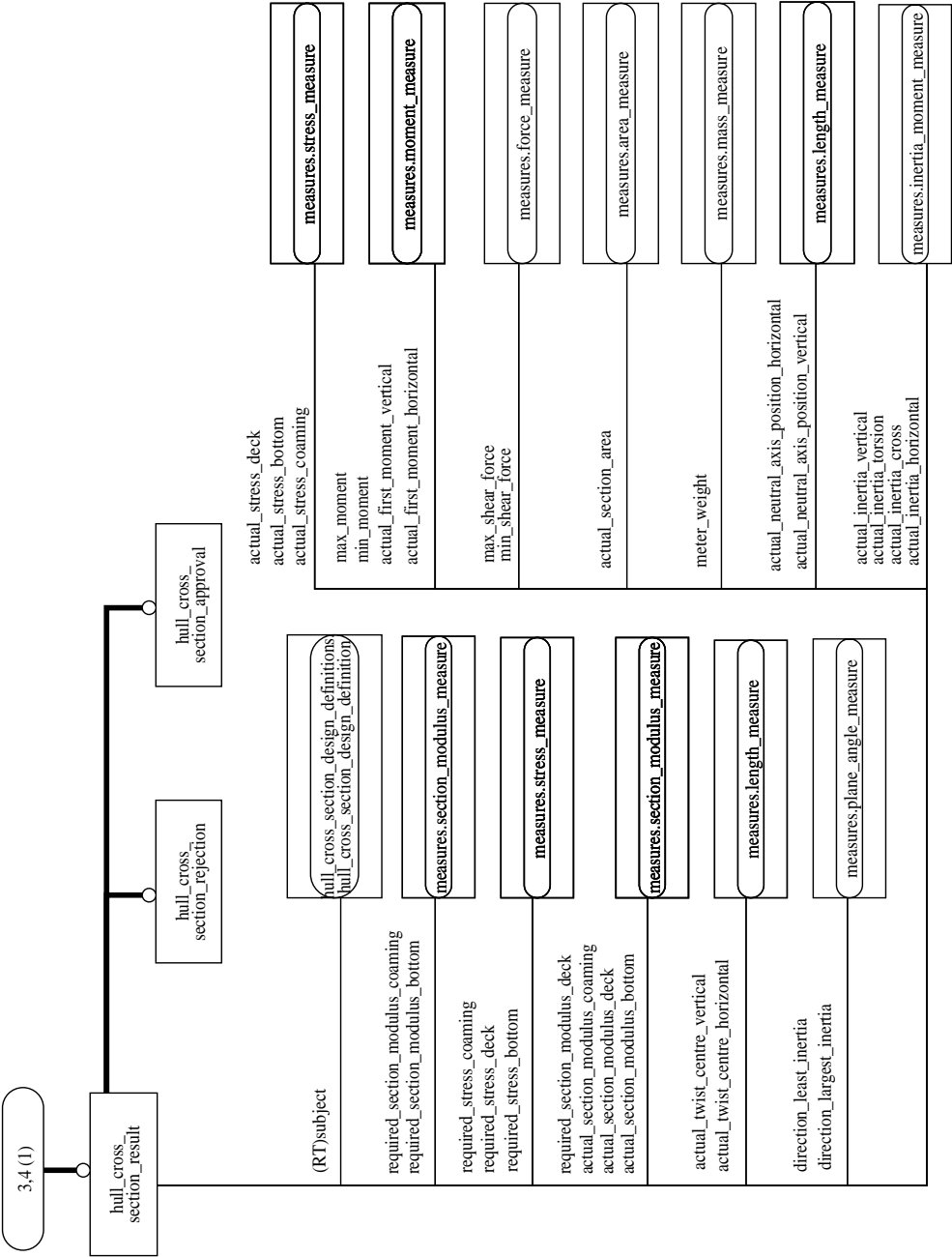


Figure G.3 – class_approvals_structure schema (3/3) in the class_approvals UoF

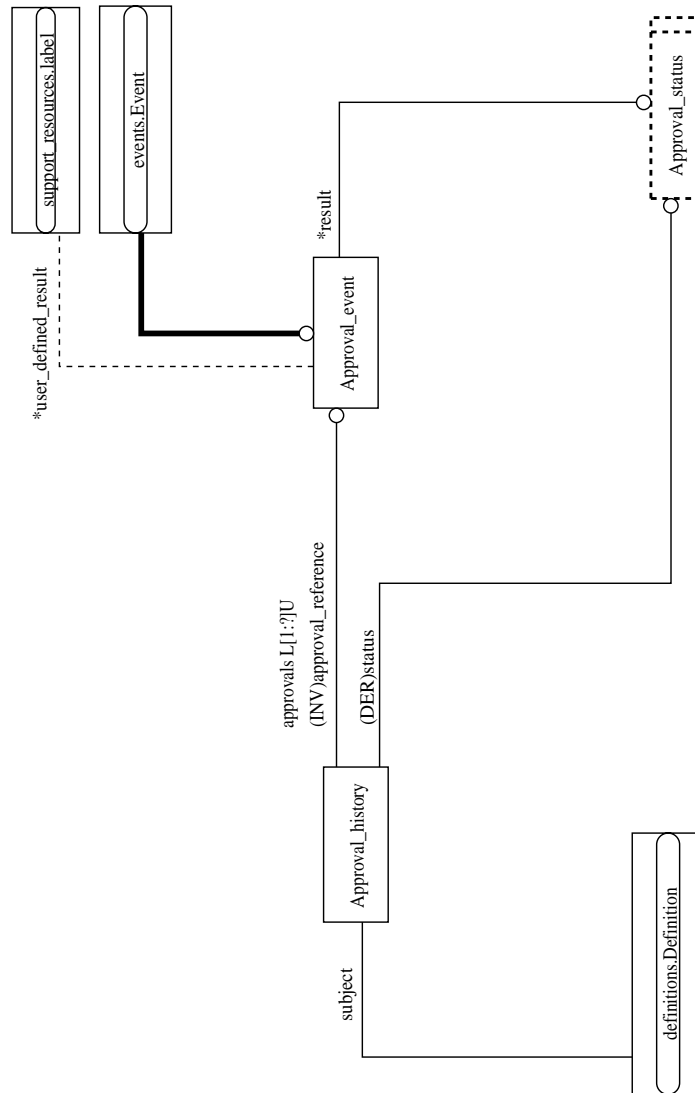


Figure G.4 – approvals schema (1/1) in the configuration_management UoF

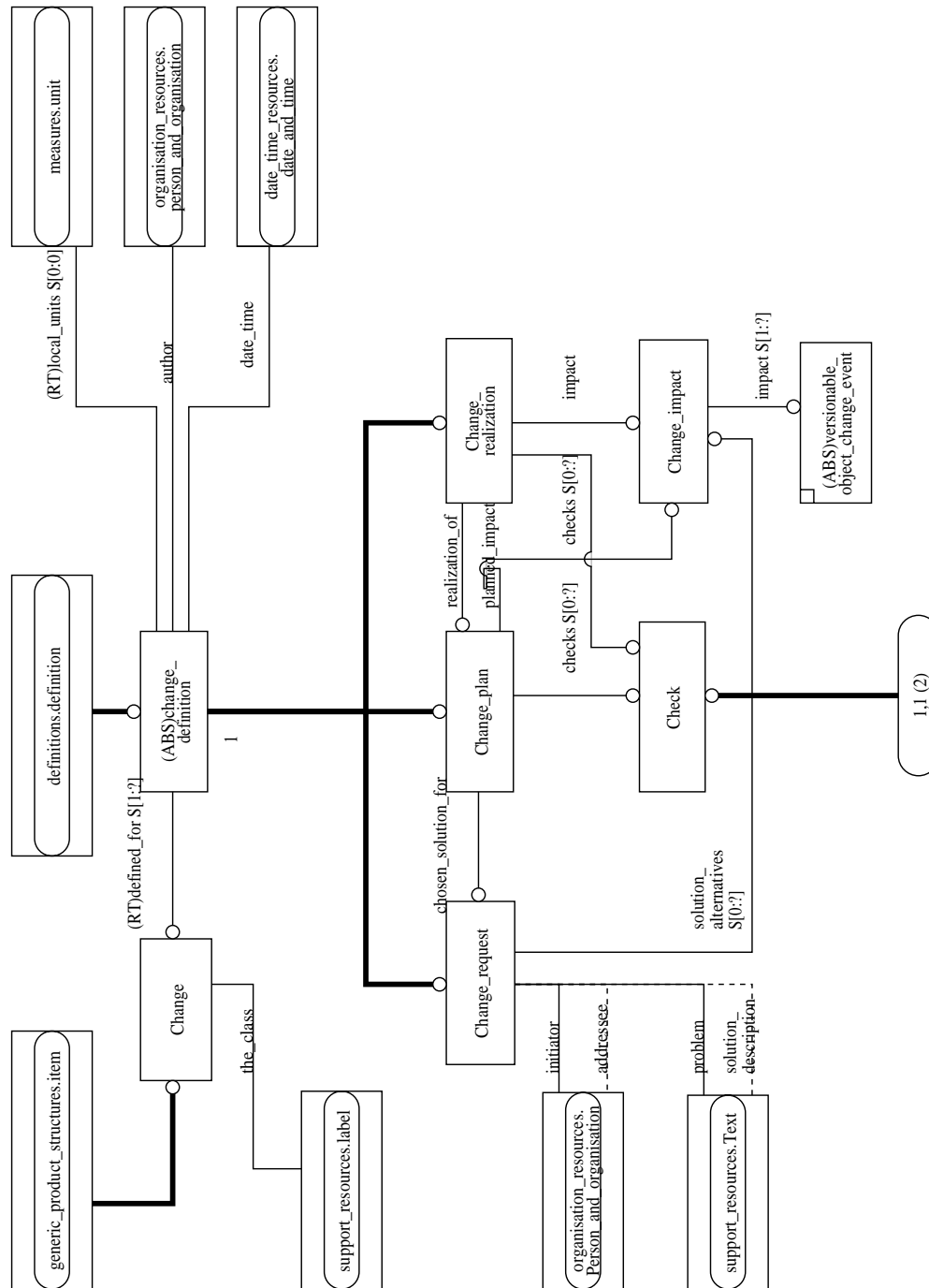


Figure G.5 – changes schema (1/2) in the configuration_management UoF

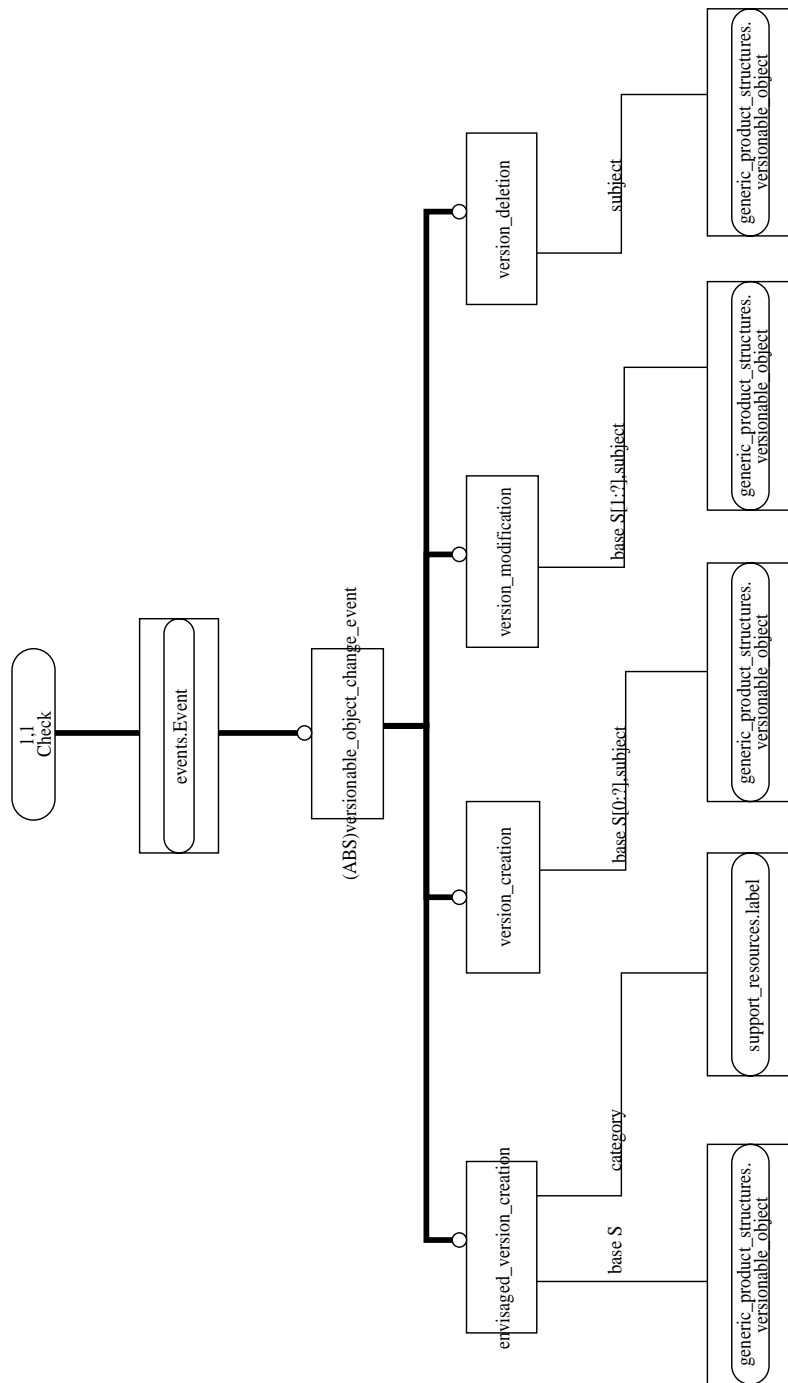


Figure G.6 – changes schema (2/2) in the configuration_management UoF

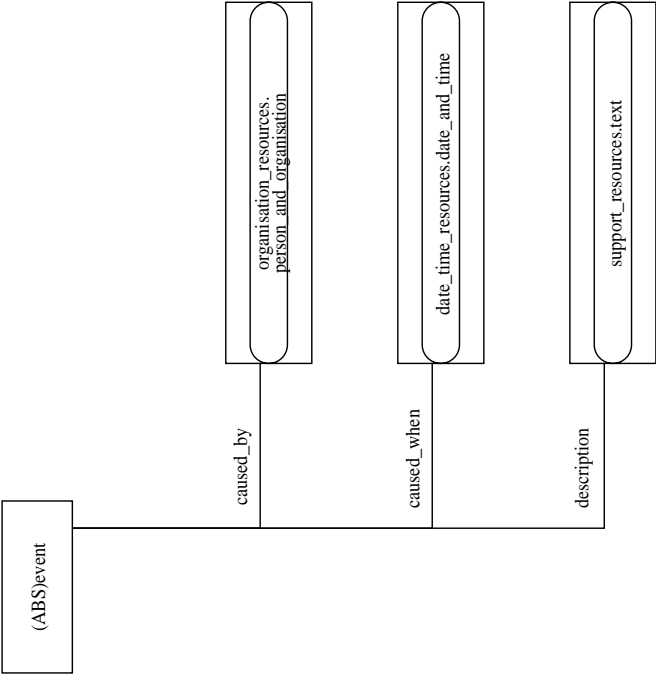


Figure G.7 – events schema (1/1) in the configuration_management UoF

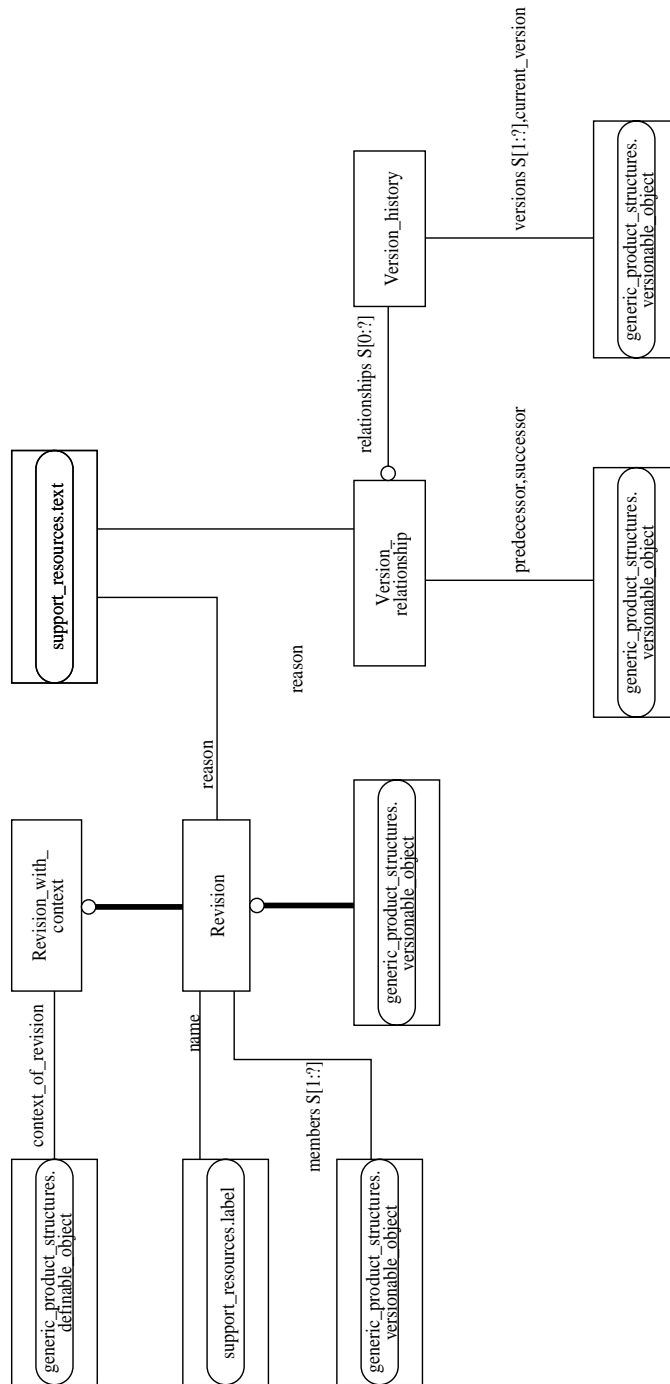


Figure G.8 – versions schema (1/1) in the configuration_management UoF

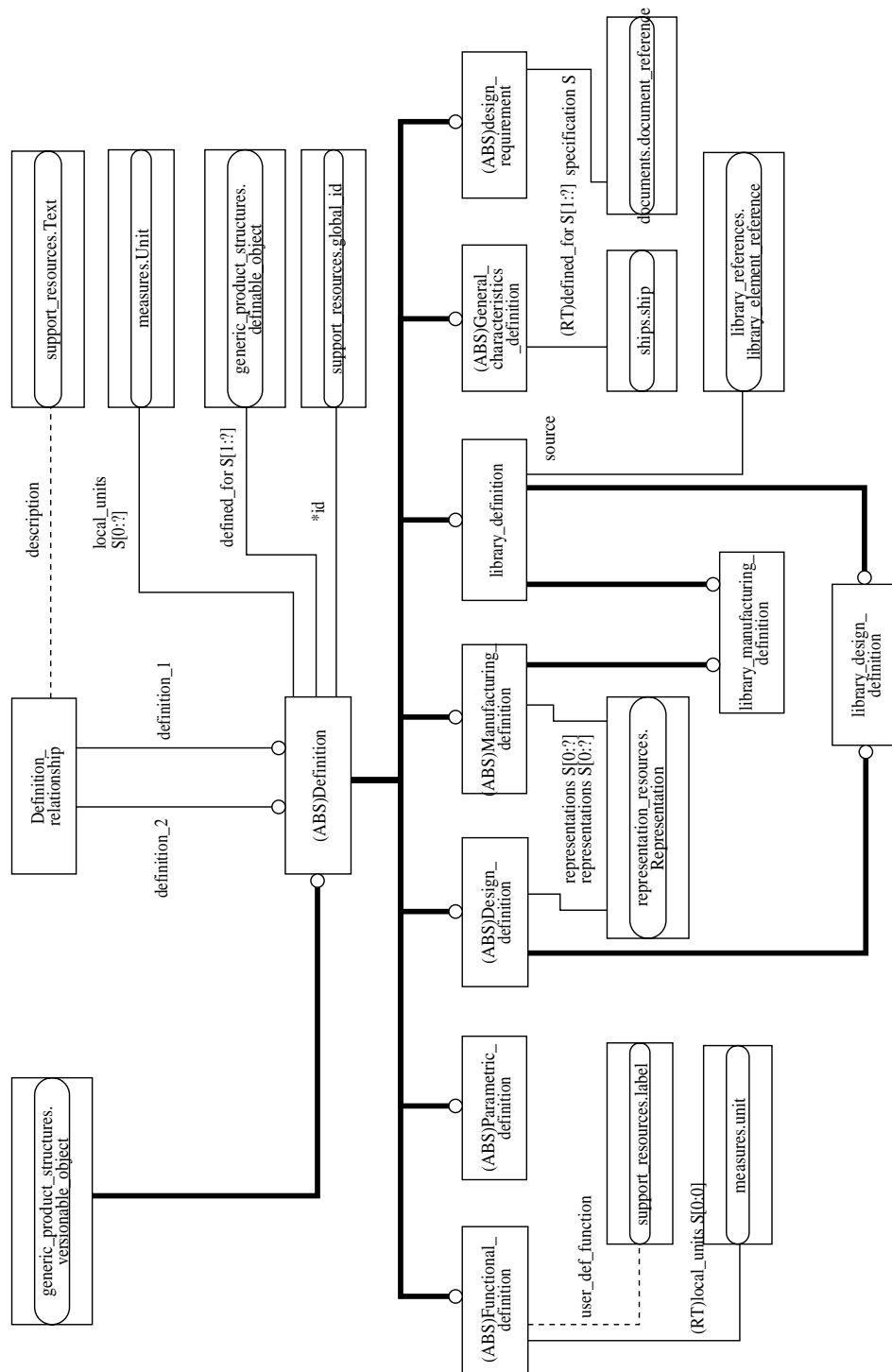


Figure G.9 – definitions schema (1/1) in the definitions UoF

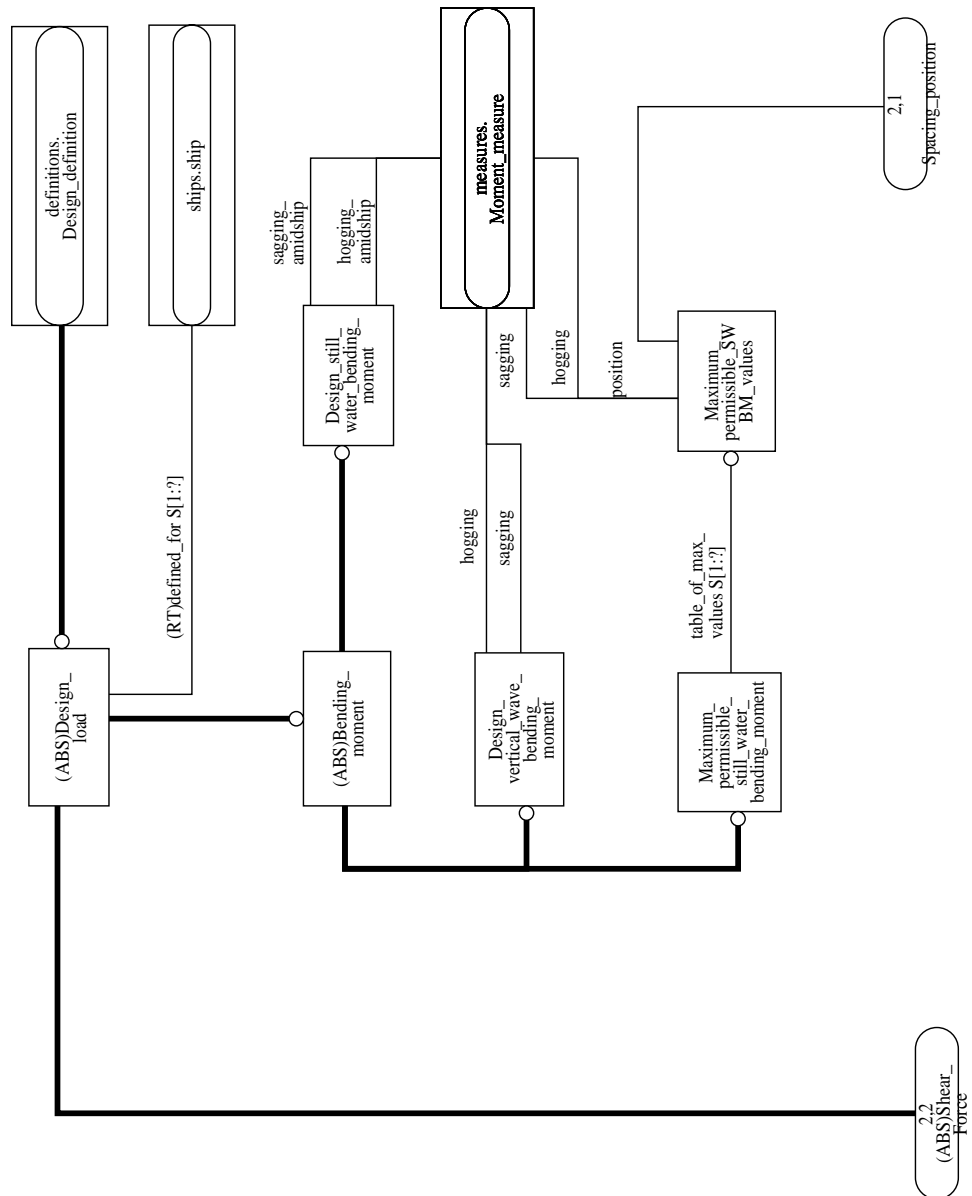
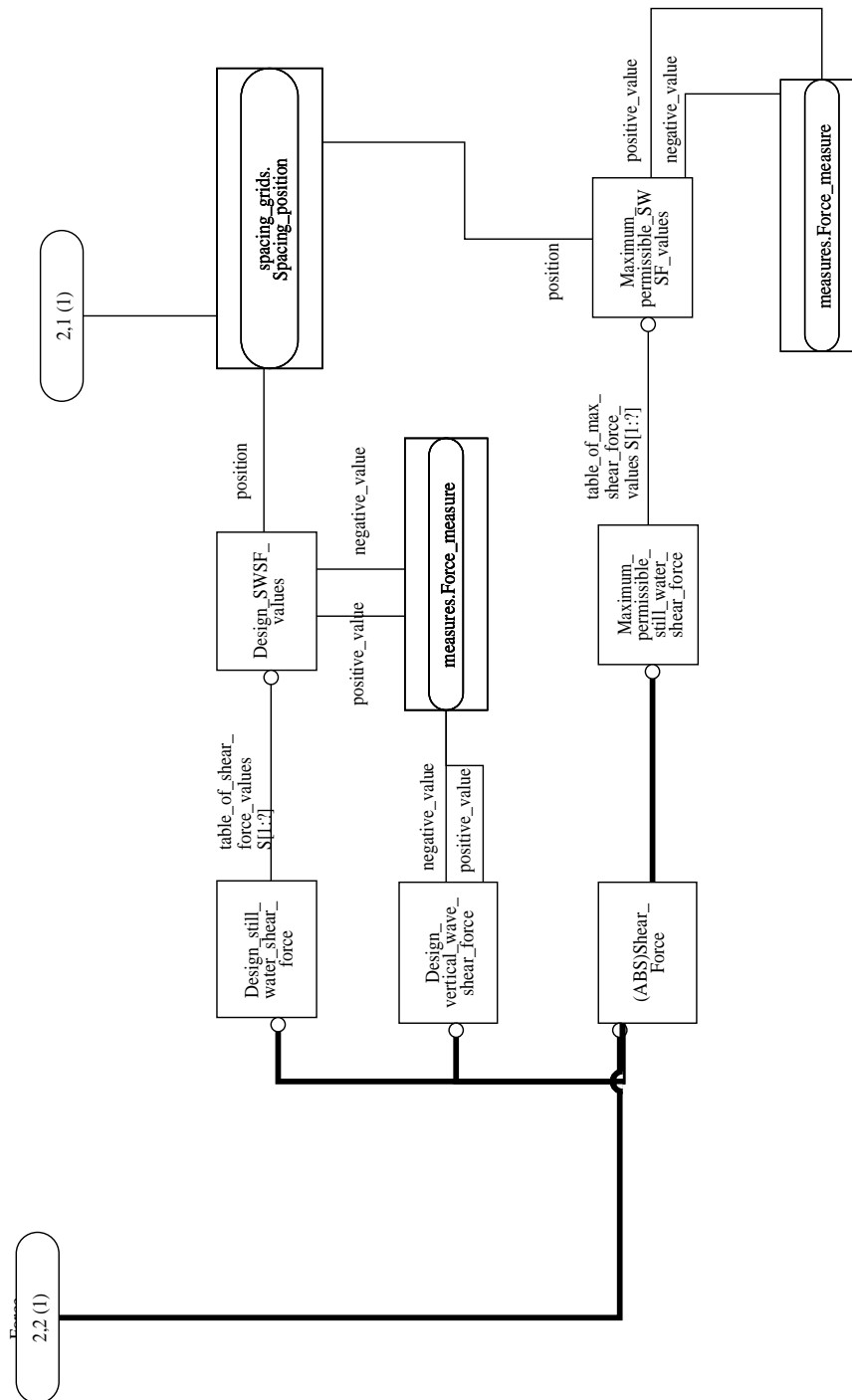


Figure G.10 – design_loads schema (1/2) in the design_loads UoF



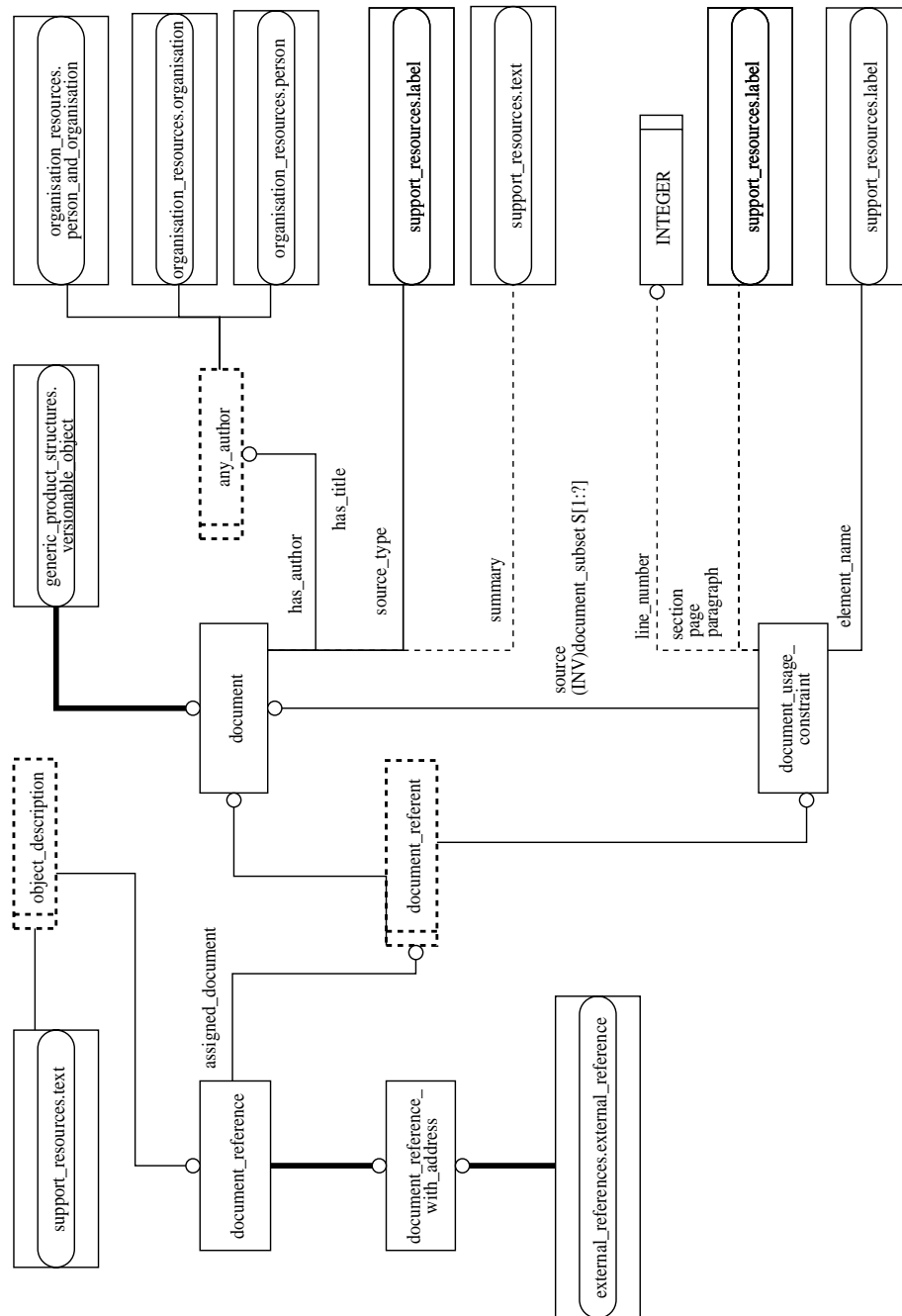


Figure G.12 – documents schema (1/1) in the external_references UoF

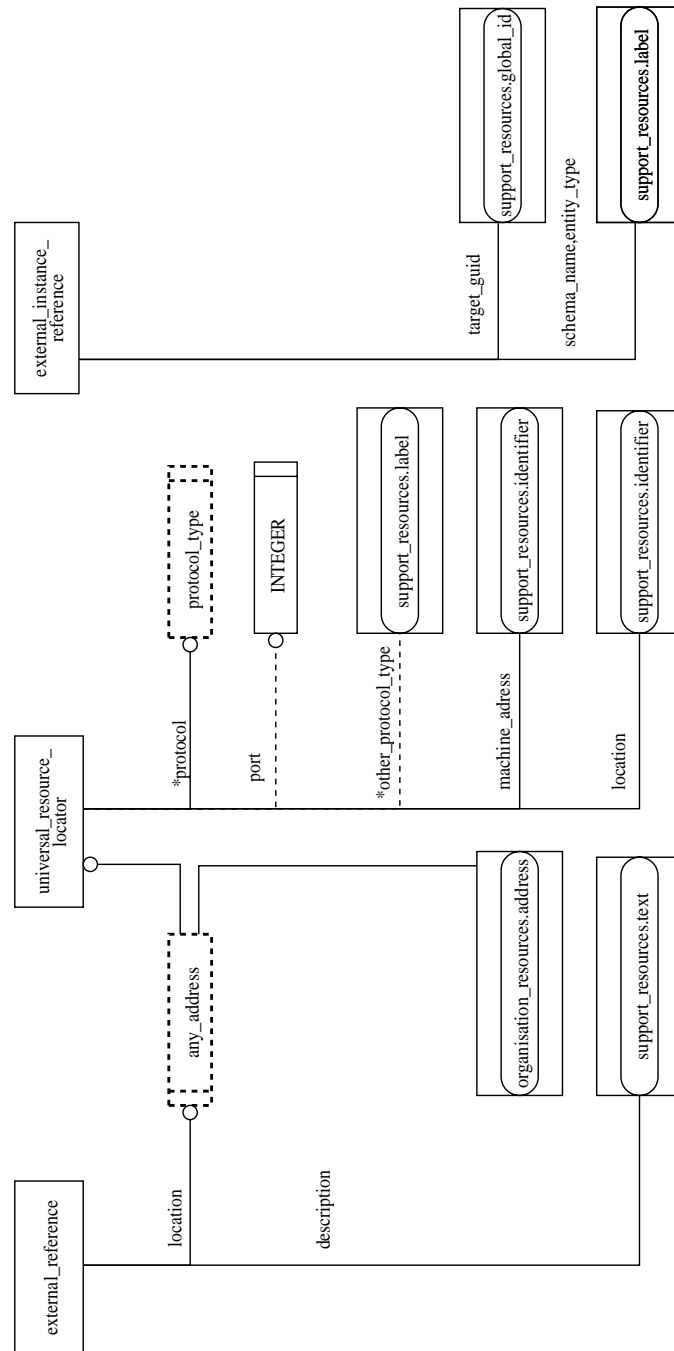


Figure G.13 – external_references schema (1/1) in the external_references UoF

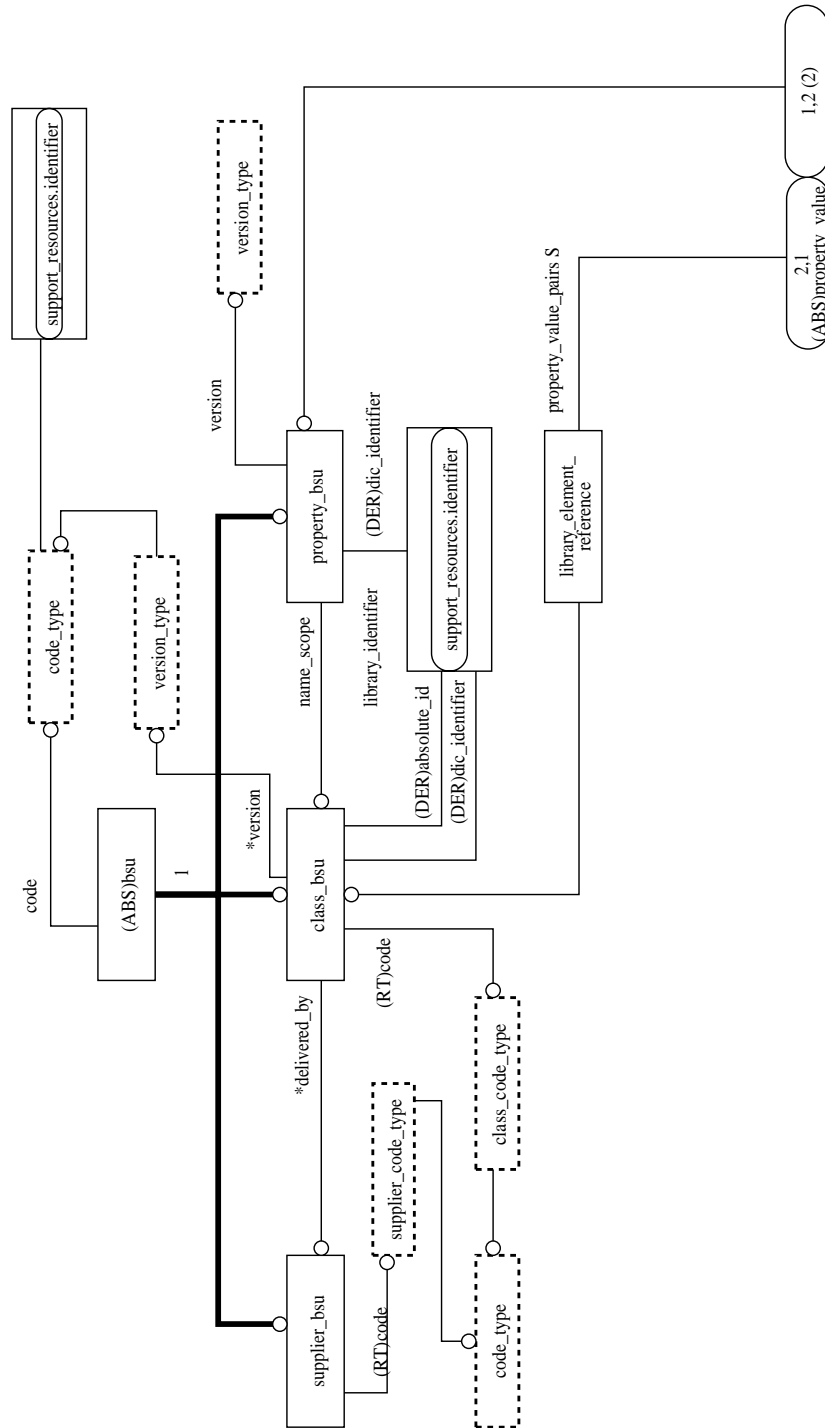


Figure G.14 – library_references schema (1/2) in the external_references UoF

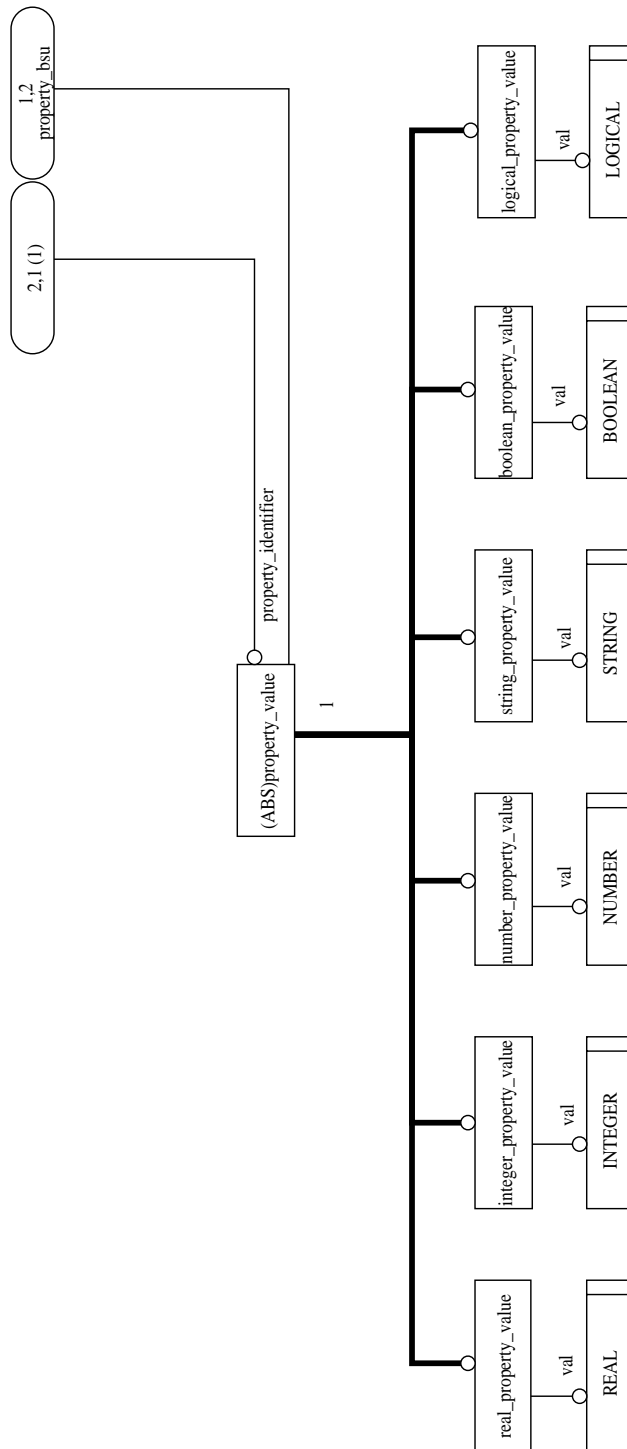


Figure G.15 – library_references schema (2/2) in the external_references UoF

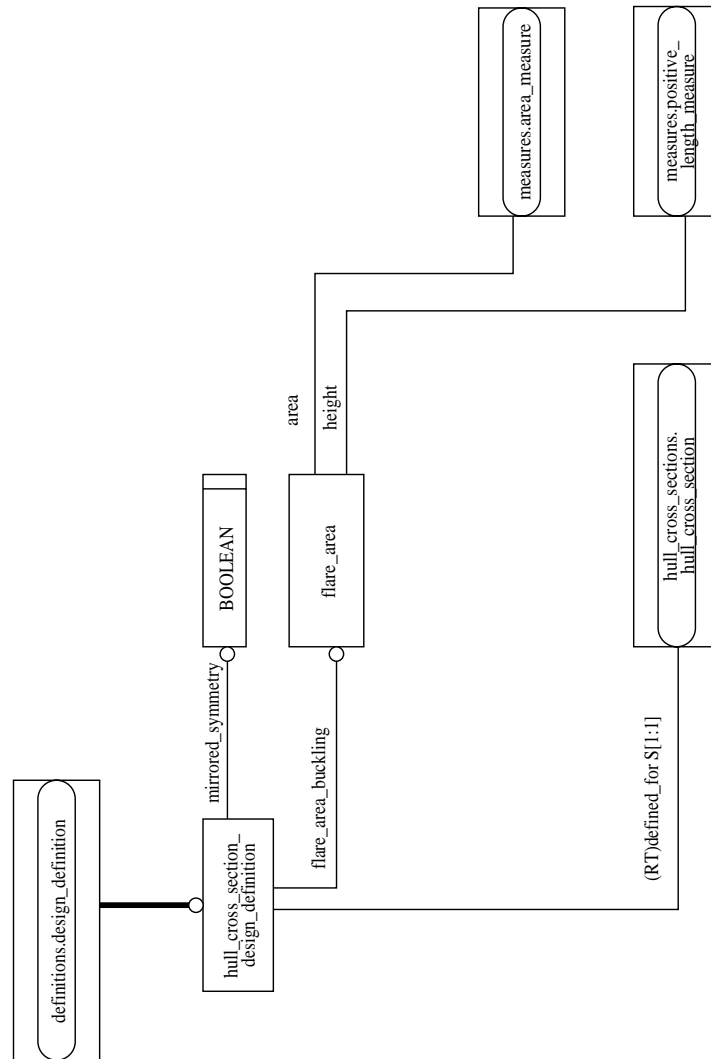


Figure G.16 – `hull_cross_section_design_definitions` schema (1/1) in the `hull_cross_sections` UoF

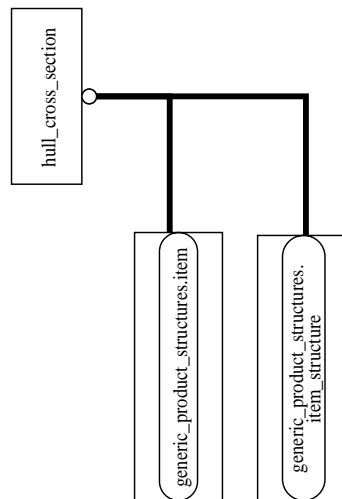


Figure G.17 – `hull_cross_sections` schema (1/1) in the `hull_cross_sections` UoF

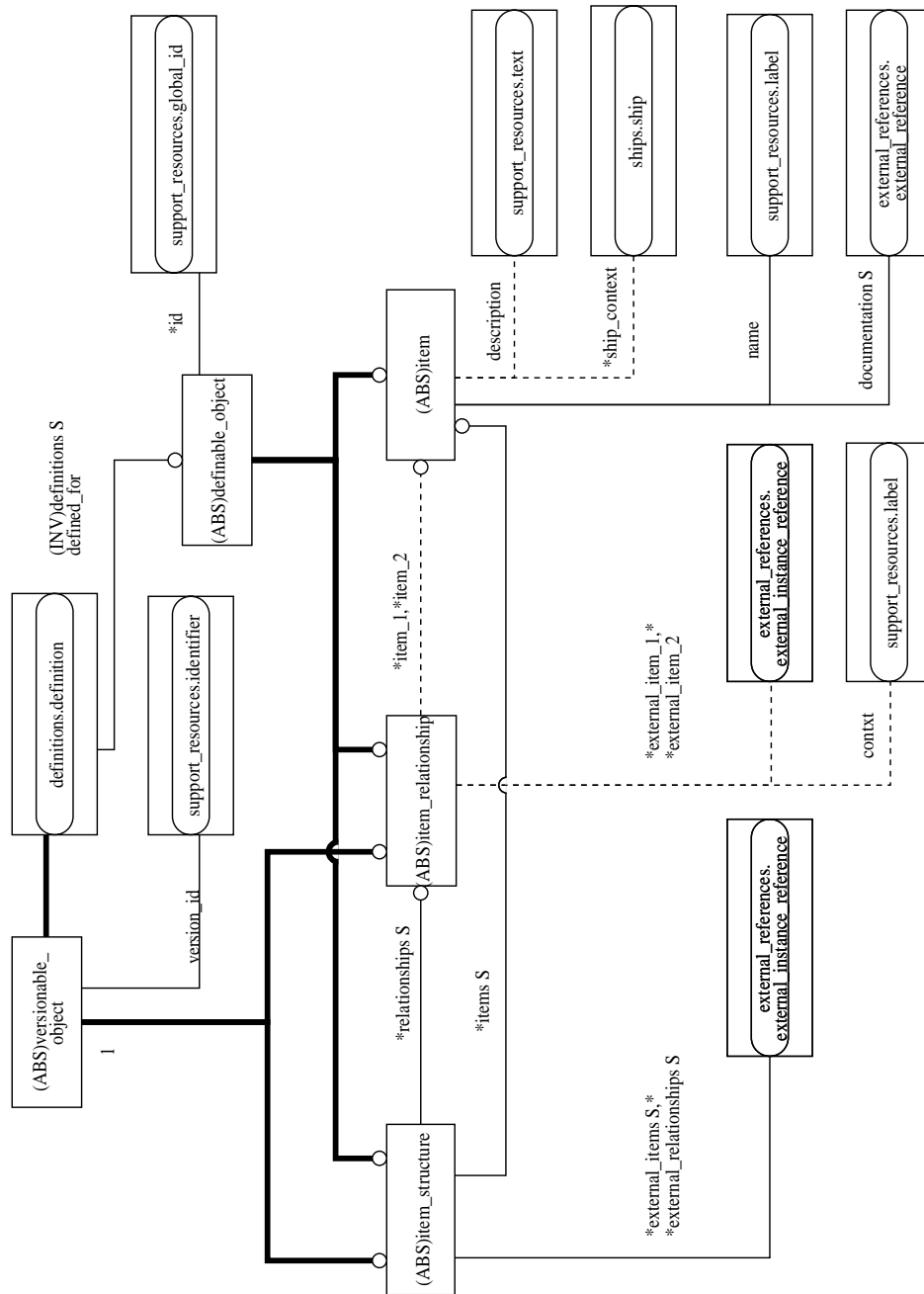


Figure G.18 – generic_product_structures schema (1/1) in the items UoF

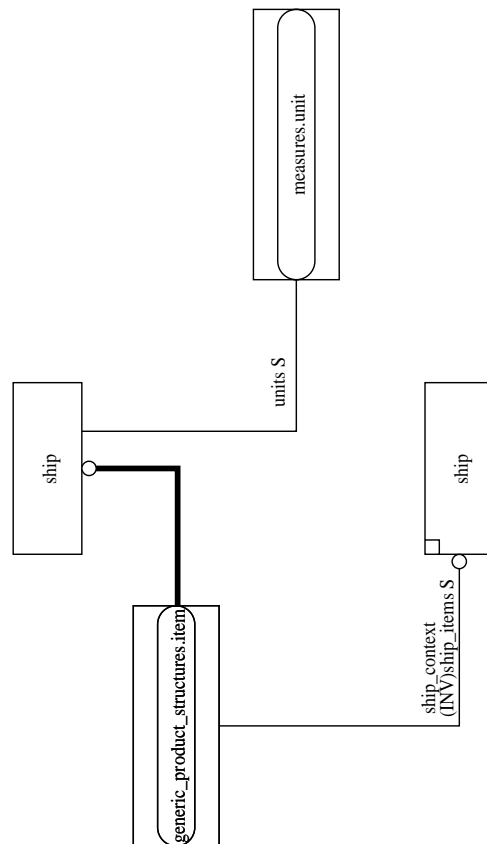


Figure G.19 – ships schema (1/1) in the items UoF

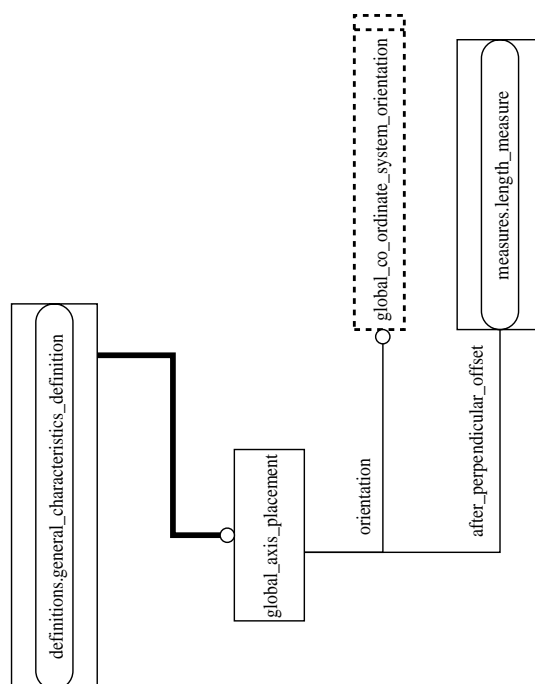


Figure G.20 – global_axis_characteristics schema (1/1) in the location_concept UoF

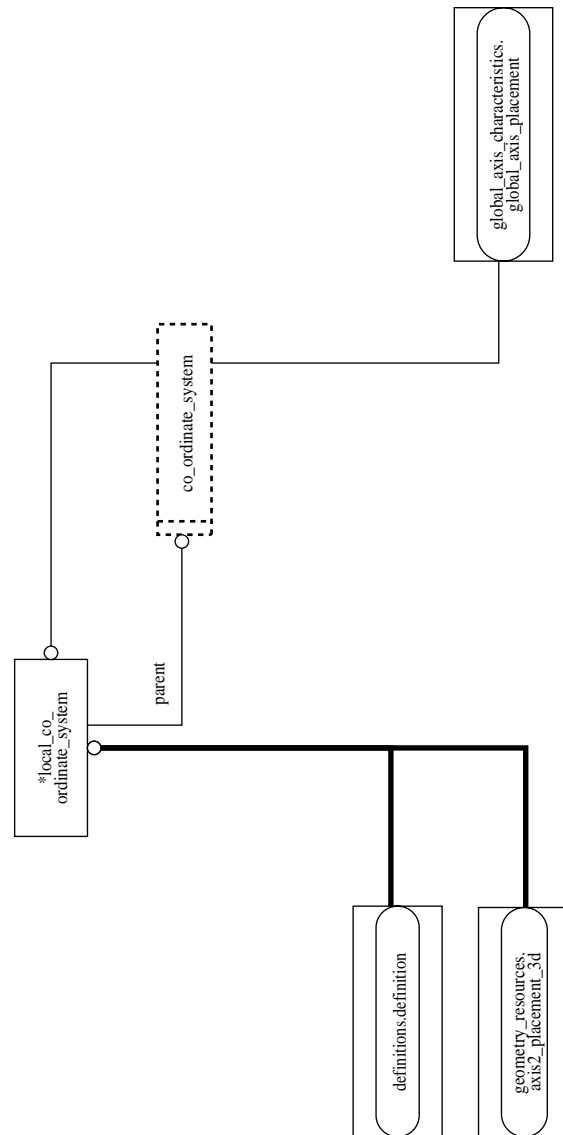


Figure G.21 – local_co_ordinate_systems schema (1/1) in the location_concept UoF

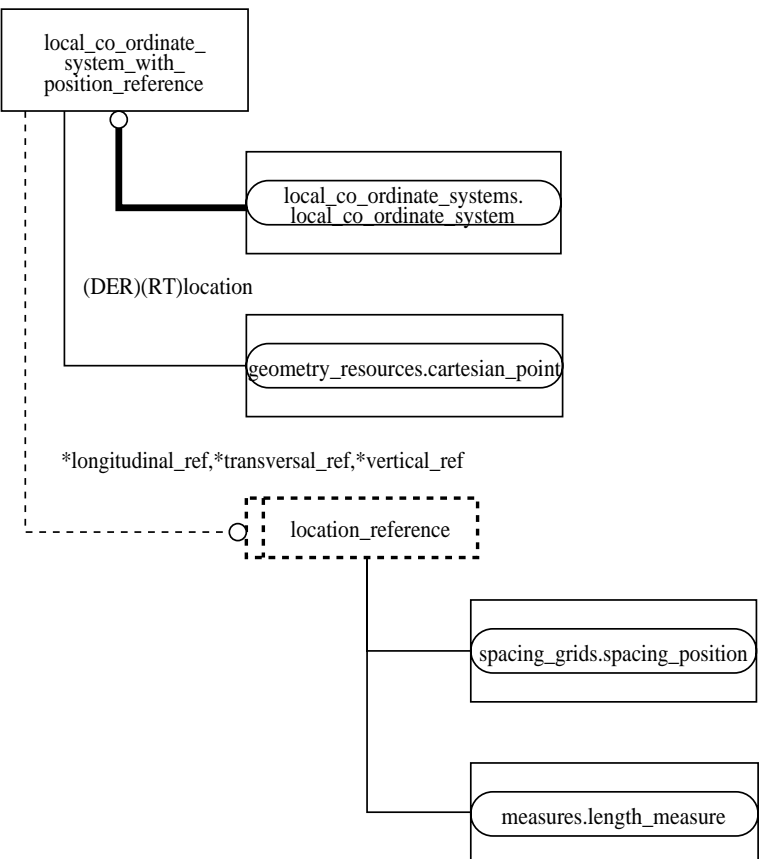


Figure G.22 – local_co_ordinate_systems_with_station_reference schema (1/1) in the location_concept UoF

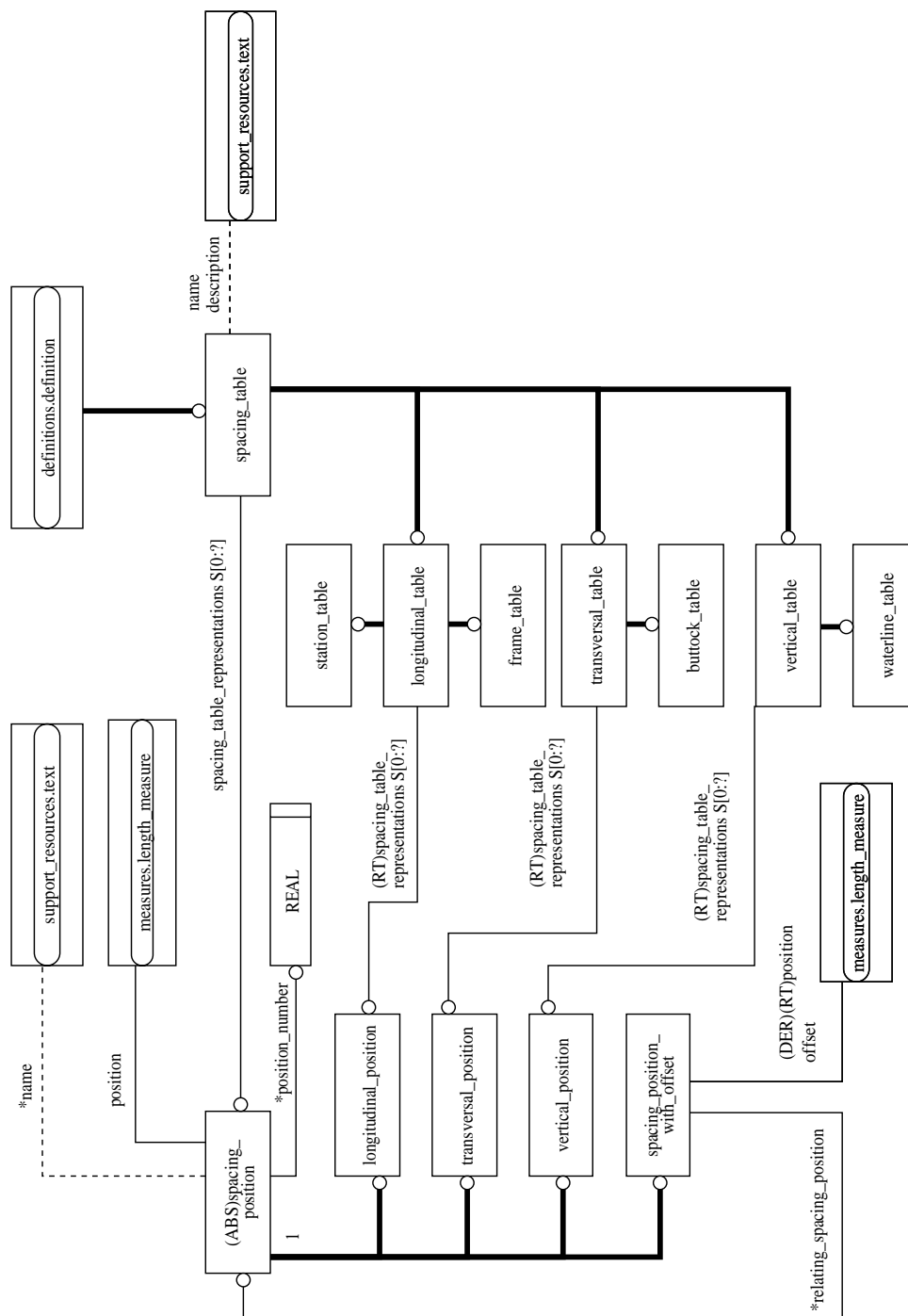


Figure G.23 – spacing_grids schema (1/1) in the location_concept UoF

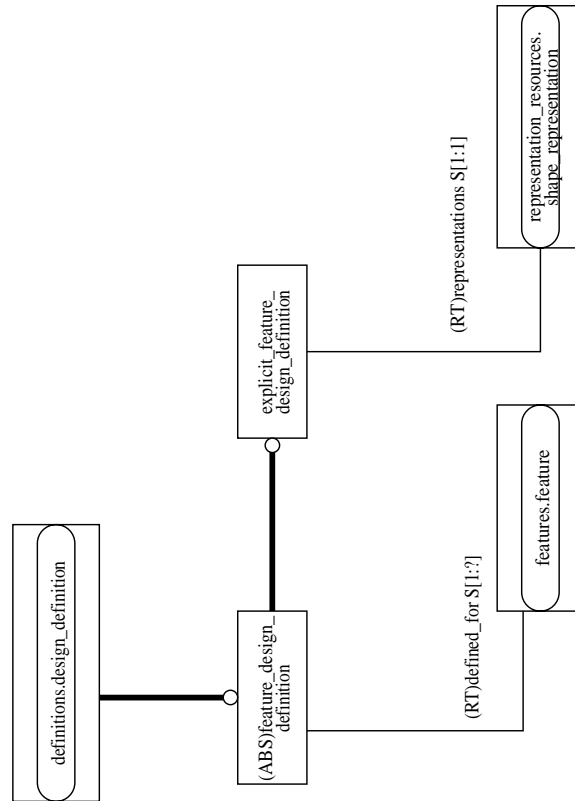


Figure G.24 – feature_design_definitions schema (1/1) in the product_structures
UoF

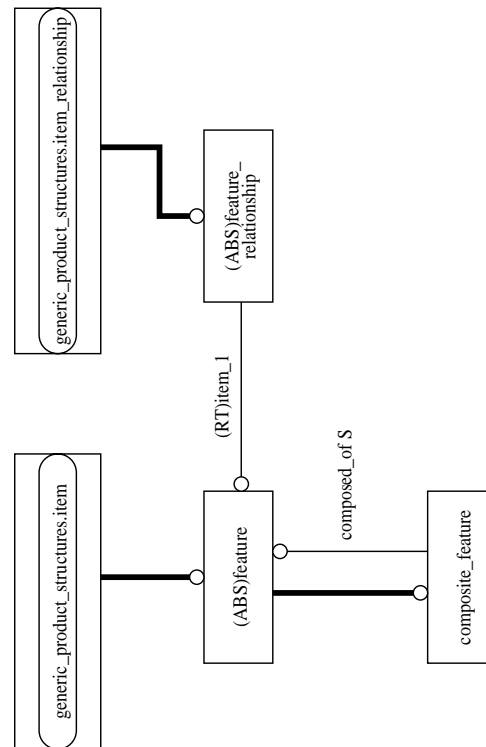


Figure G.25 – features schema (1/1) in the product_structures UoF

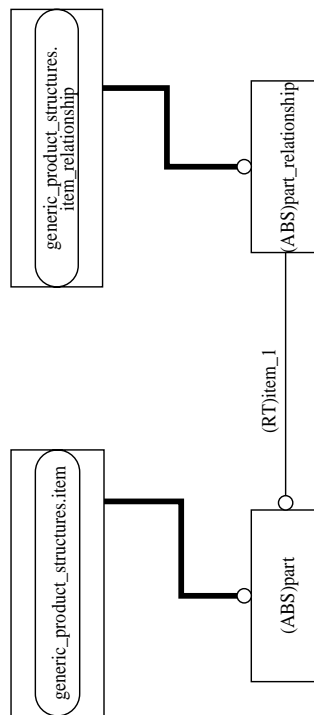


Figure G.26 – parts schema (1/1) in the product_structures UoF

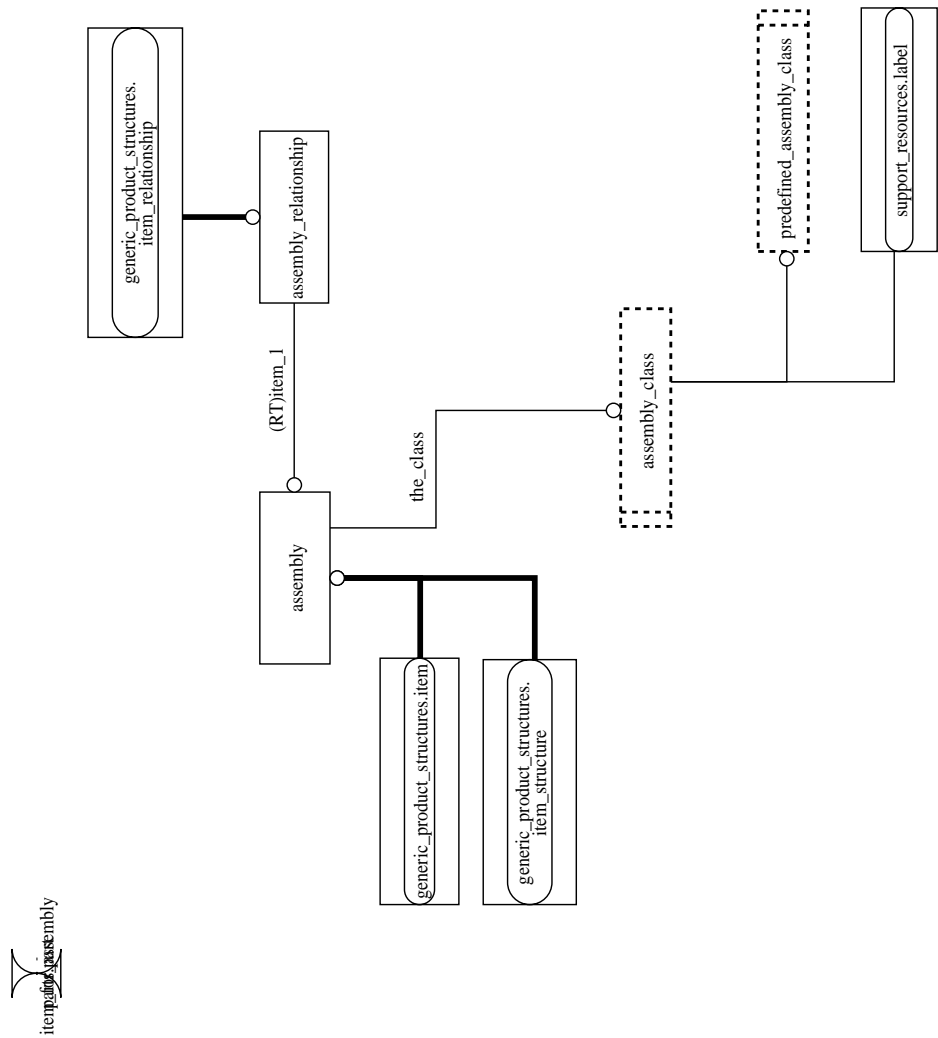


Figure G.27 – product_structure_by_assembly schema (1/1) in the product_structures UoF

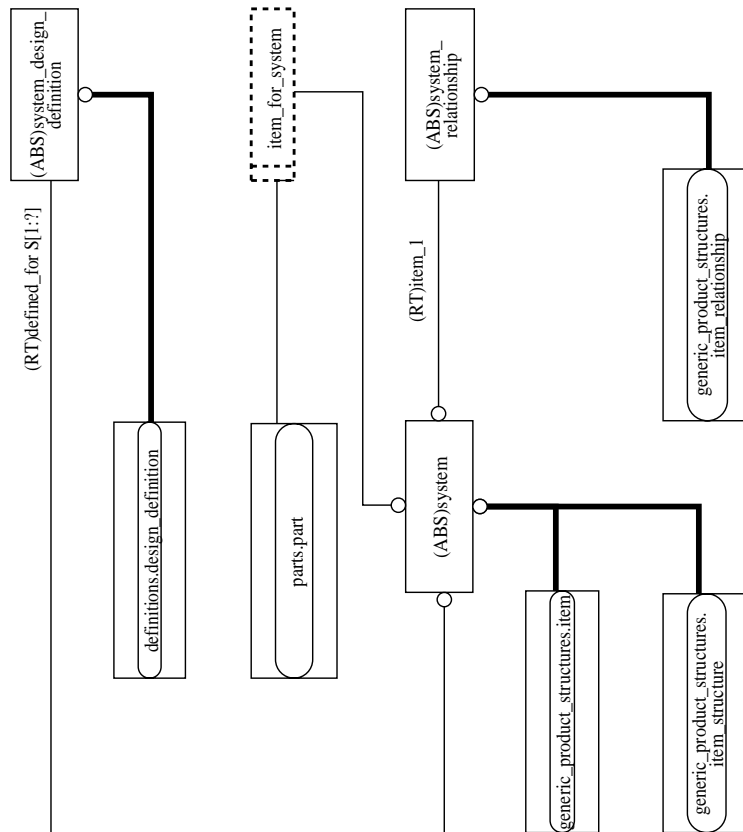


Figure G.28 – product_structure_by_system schema (1/1) in the product_structures UoF

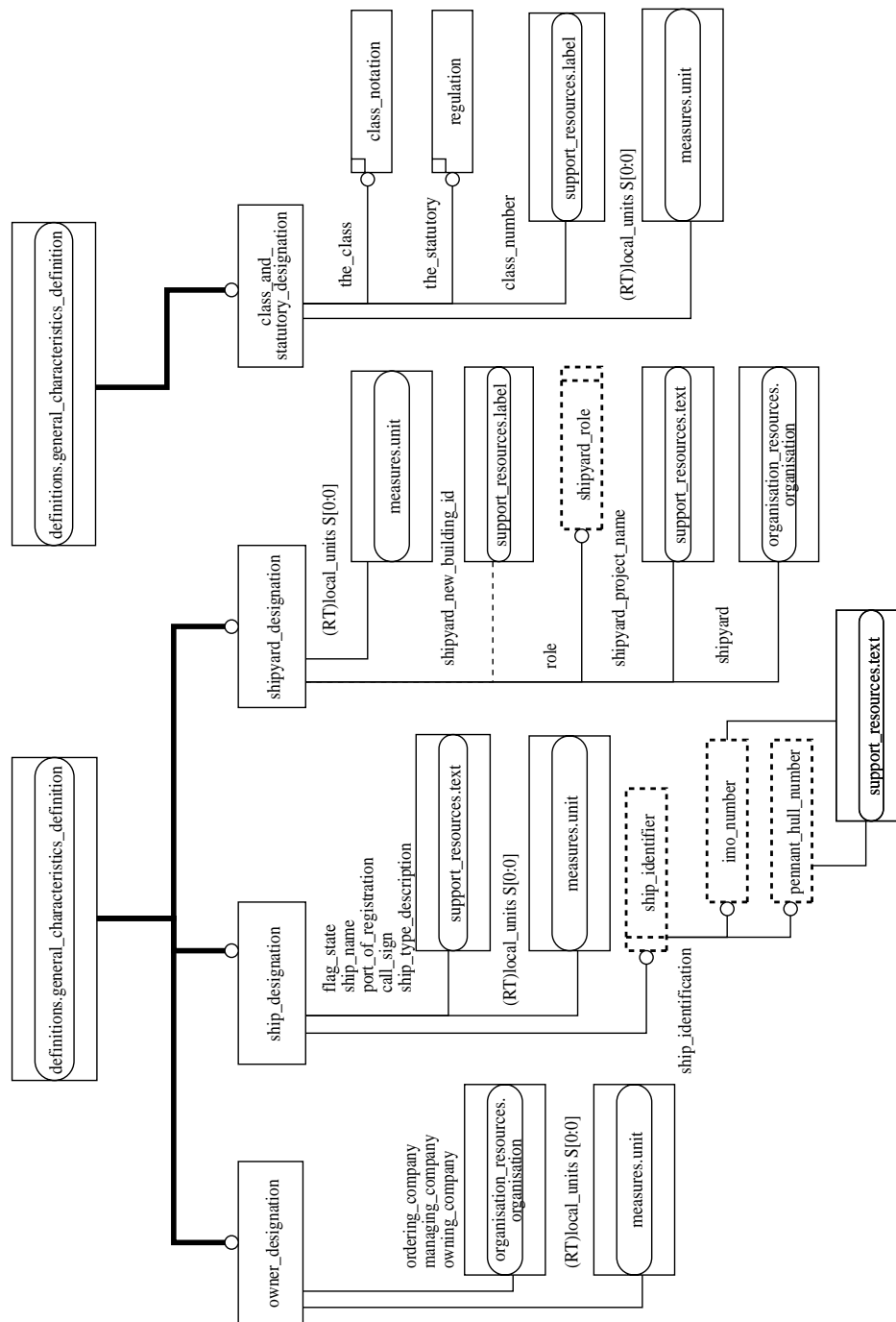


Figure G.29 – designation_characteristics schema (1/2) in the ship_general_characteristics UoF

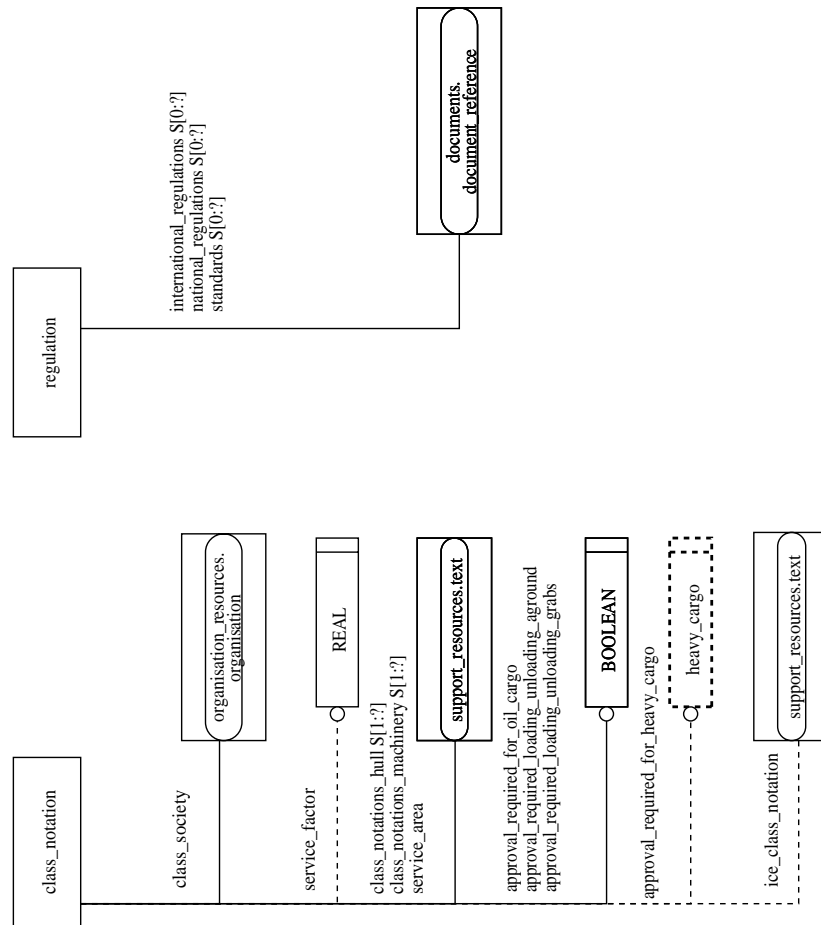


Figure G.30 – designation_characteristics schema (2/2) in the ship_general_characteristics UoF

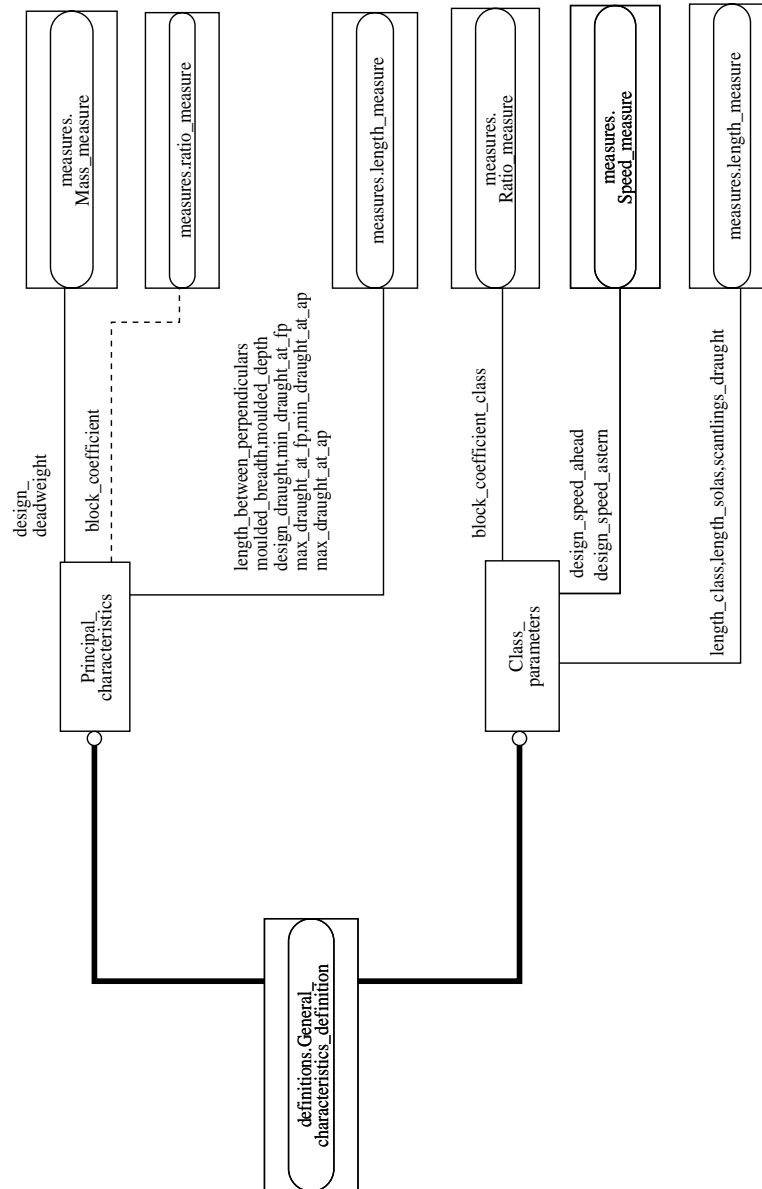


Figure G.31 – dimension_characteristics schema (1/1) in the ship_general_characteristics UoF

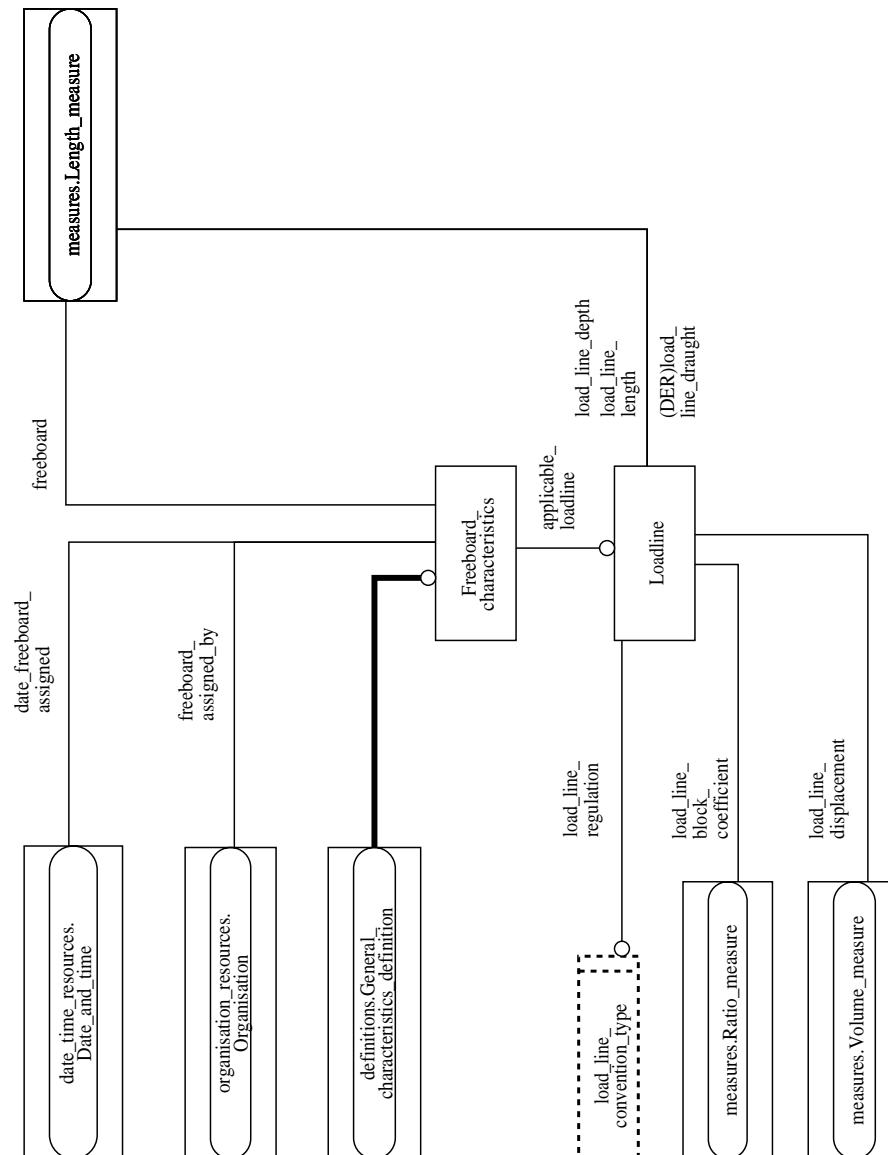


Figure G.32 – freeboard_characteristics schema (1/1) in the ship_general_characteristics UoF

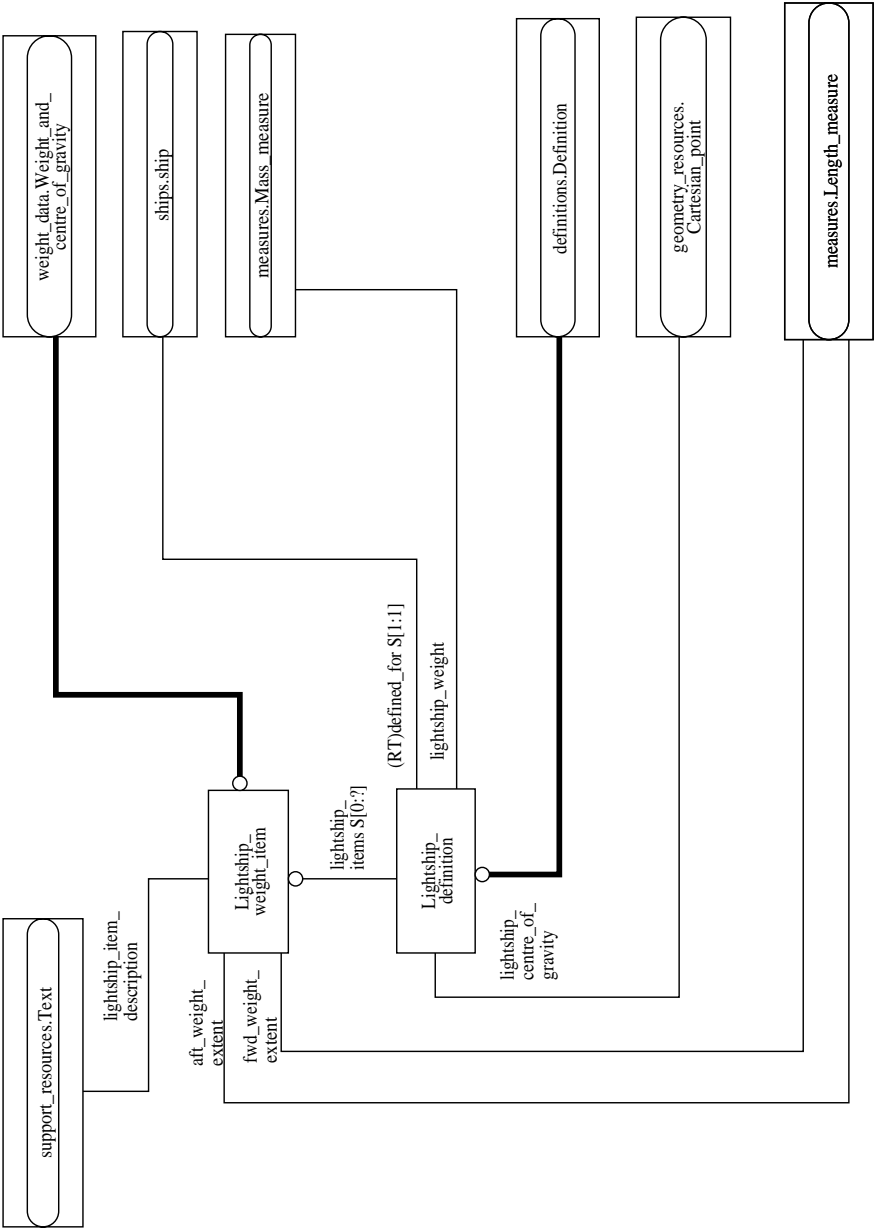


Figure G.33 – lightship_weight schema (1/1) in the ship_general_characteristics UoF

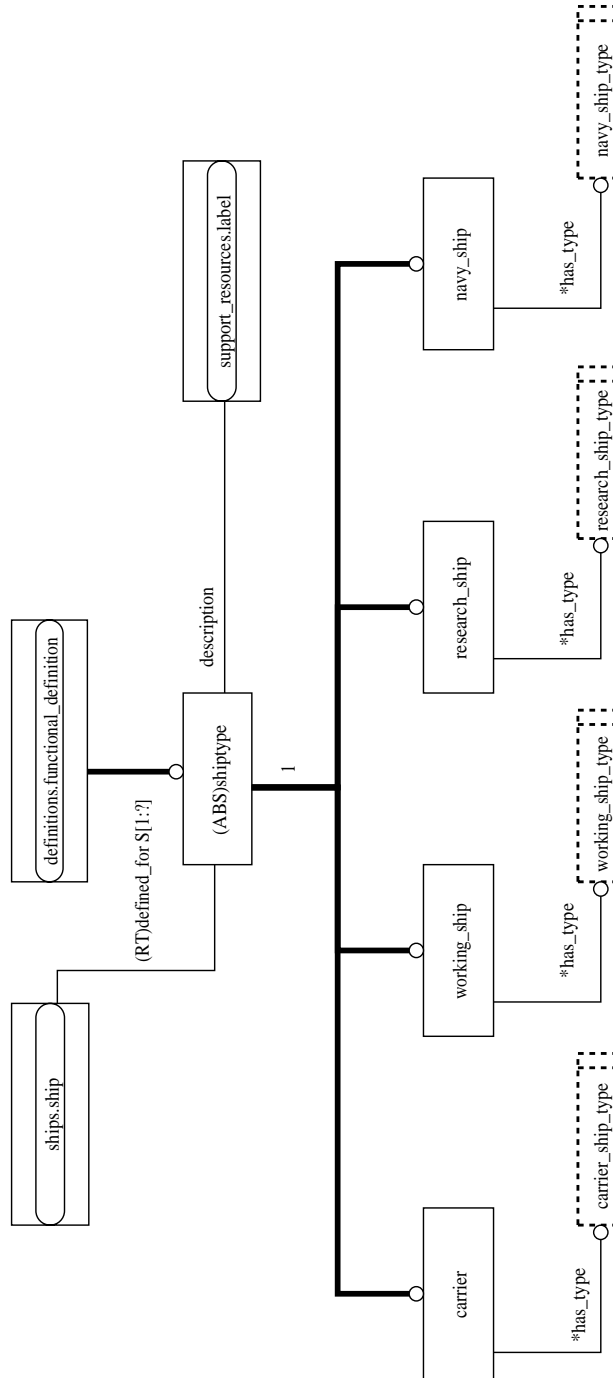


Figure G.34 – ship_types schema (1/1) in the product_definitions UoF

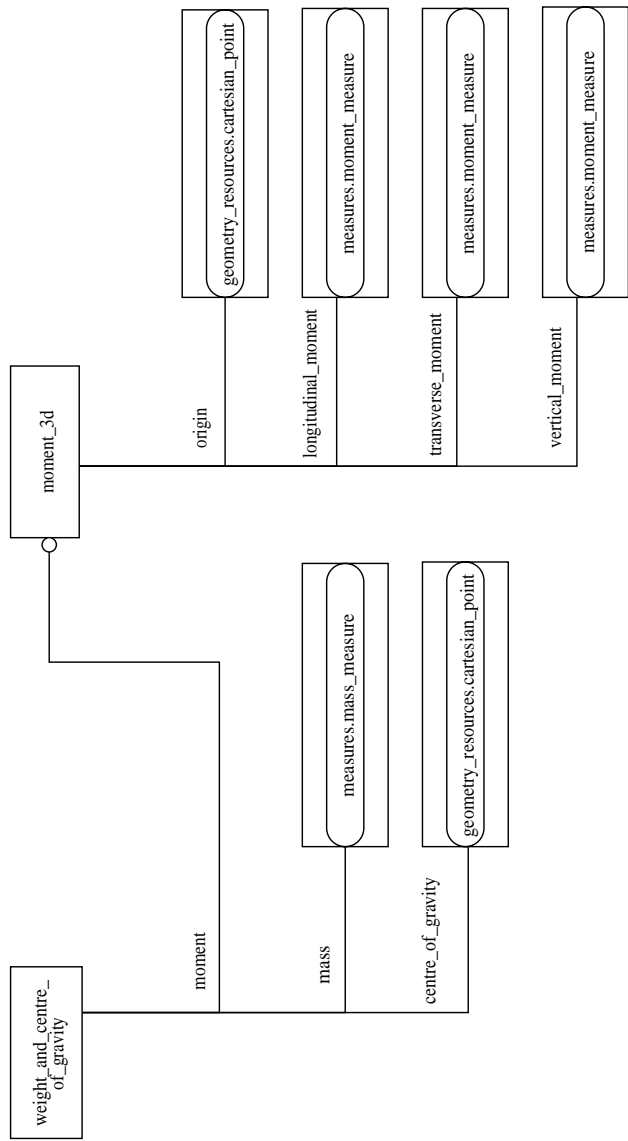


Figure G.35 – weight_data schema (1/1) in the ship_general_characteristics UoF

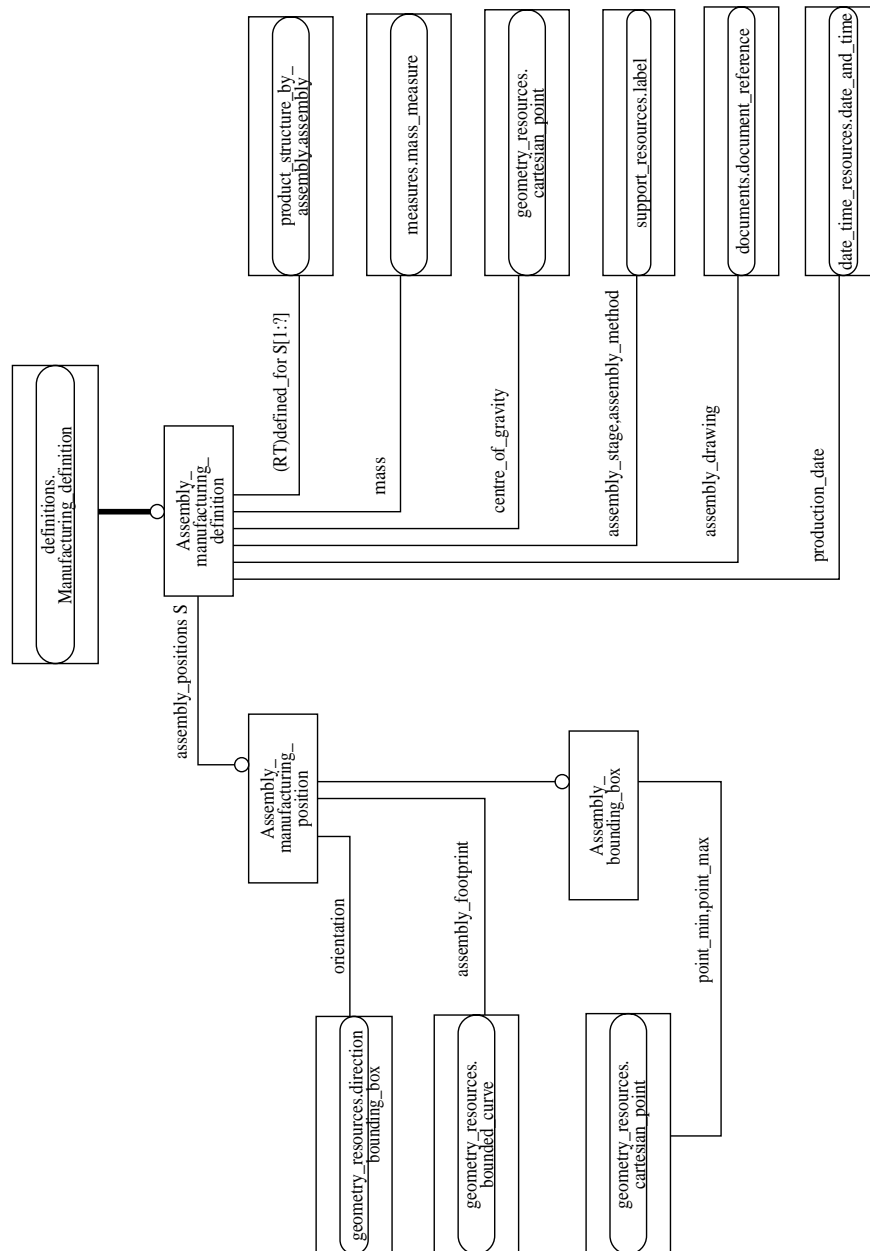


Figure G.36 – assembly_manufacturing_definitions schema (1/1) in the ship_manufacturing_definitions UoF

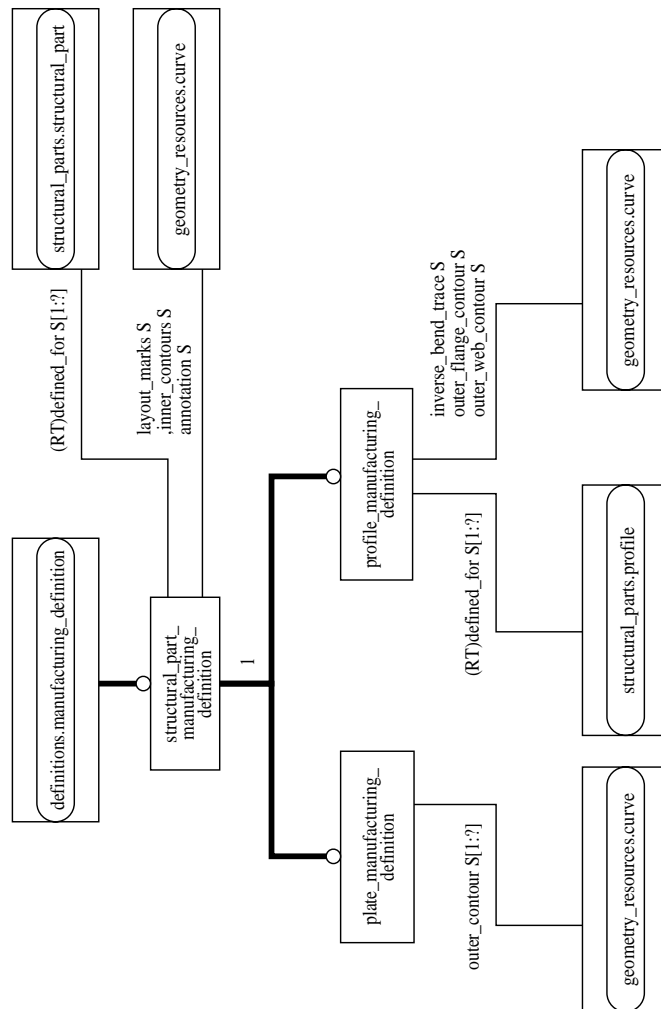


Figure G.37 – structural_part_manufacturing_definitions schema (1/1) in the ship_manufacturing_definitions UoF

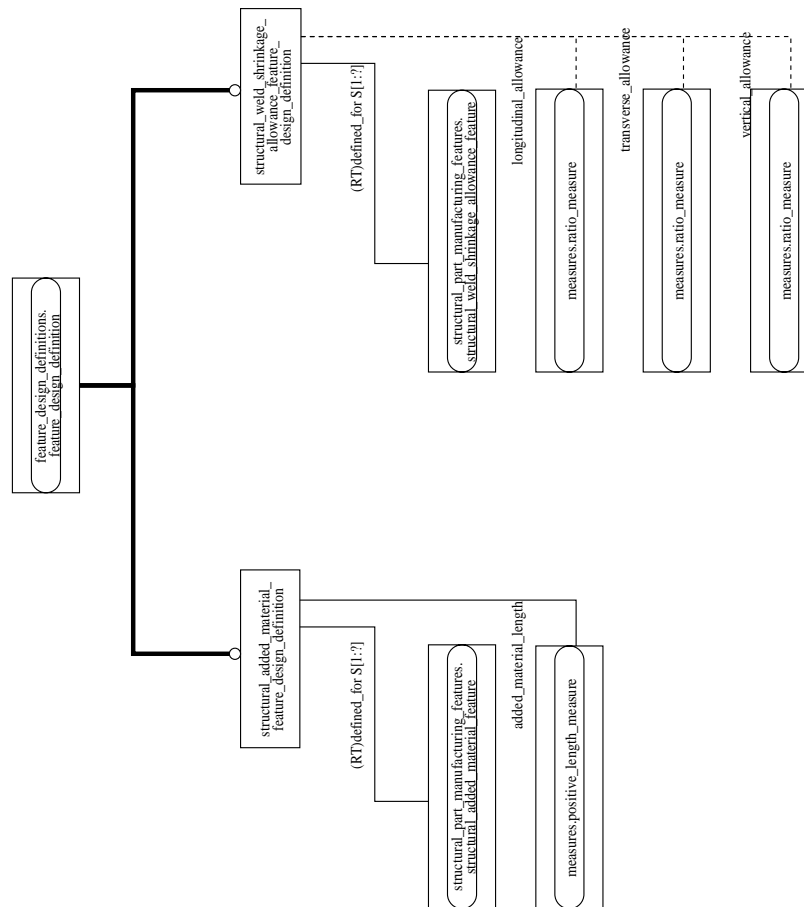


Figure G.38 – structural_part_manufacturing_features_design_definitions schema
(1/1) in the ship_manufacturing_definitions UoF

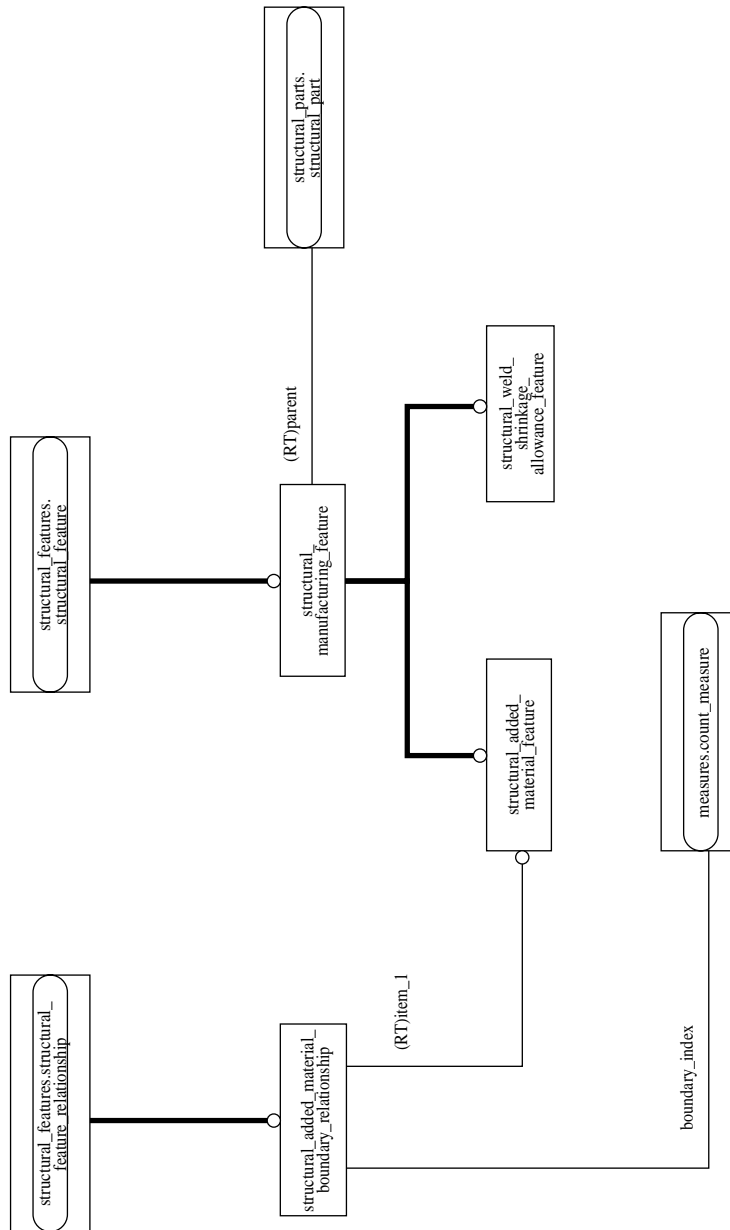


Figure G.39 – structural_part_manufacturing_features schema (1/1) in the ship_manufacturing_definitions UoF

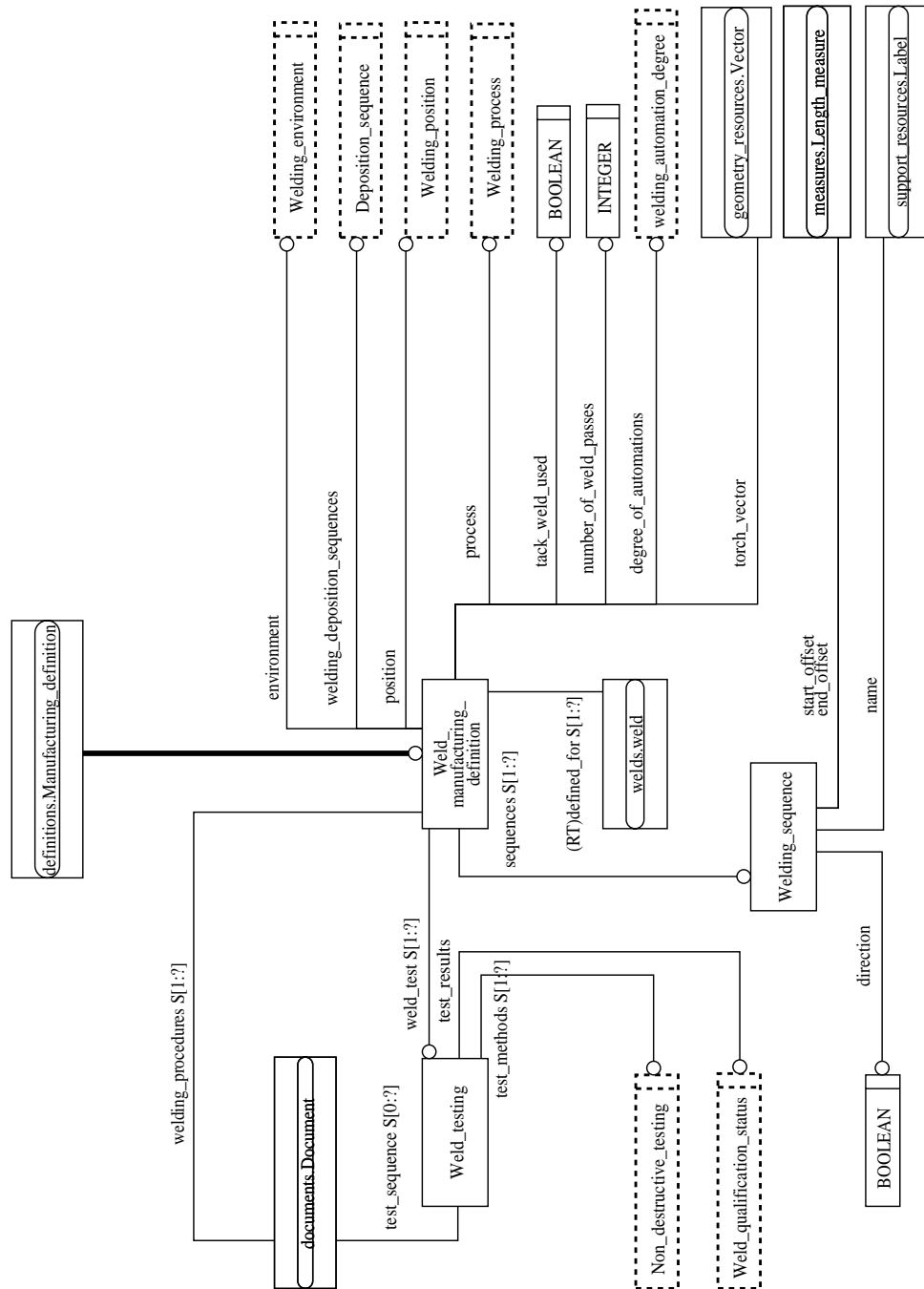


Figure G.40 – weld_manufacturing_definitions schema (1/1) in the ship_manufacturing_definitions UoF

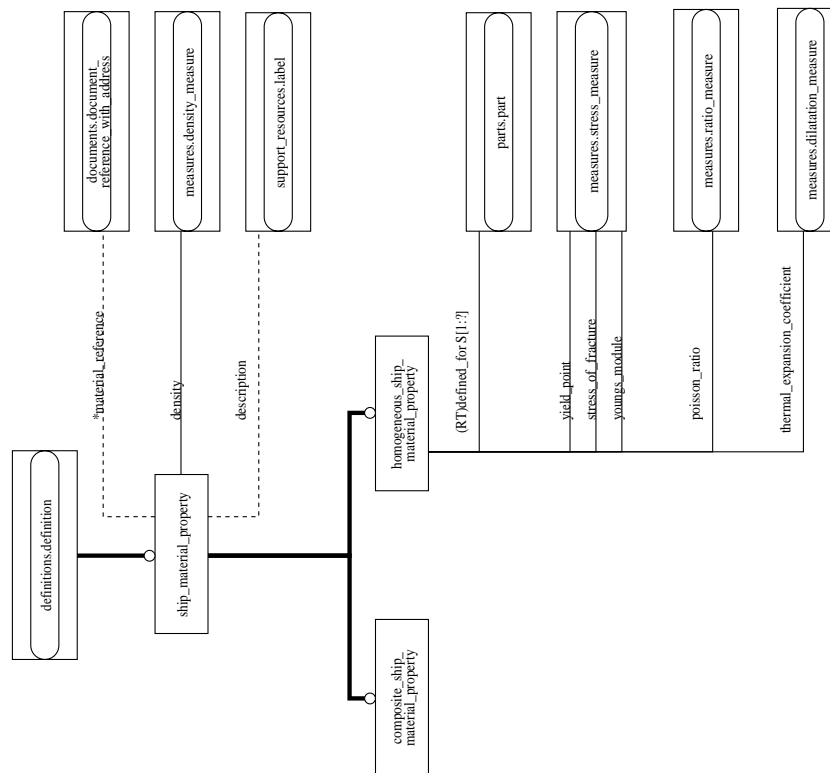


Figure G.41 – ship_material_properties schema (1/1) in the ship_materials UoF

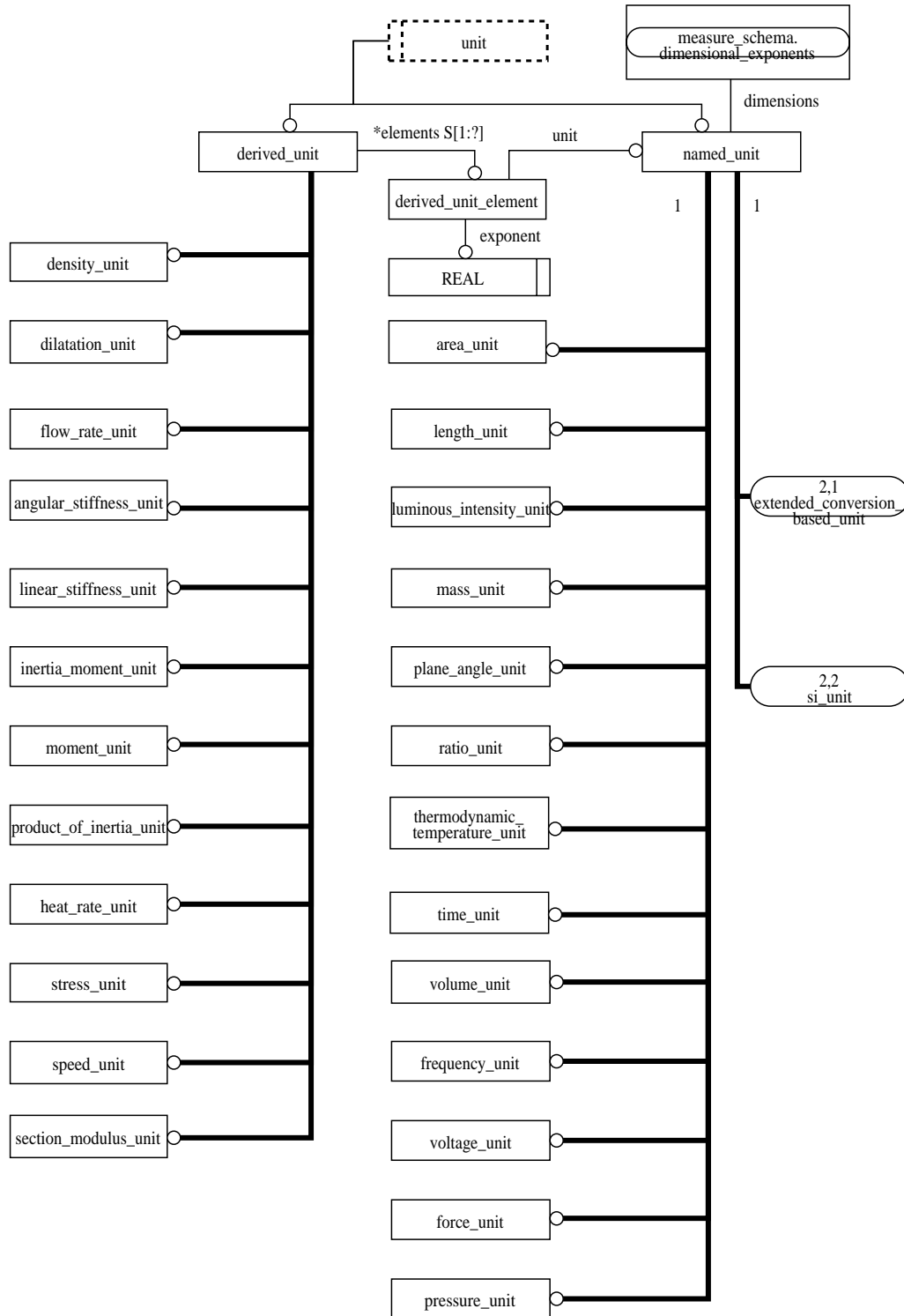


Figure G.42 – measures schema (1/4) in the ship_measures UoF

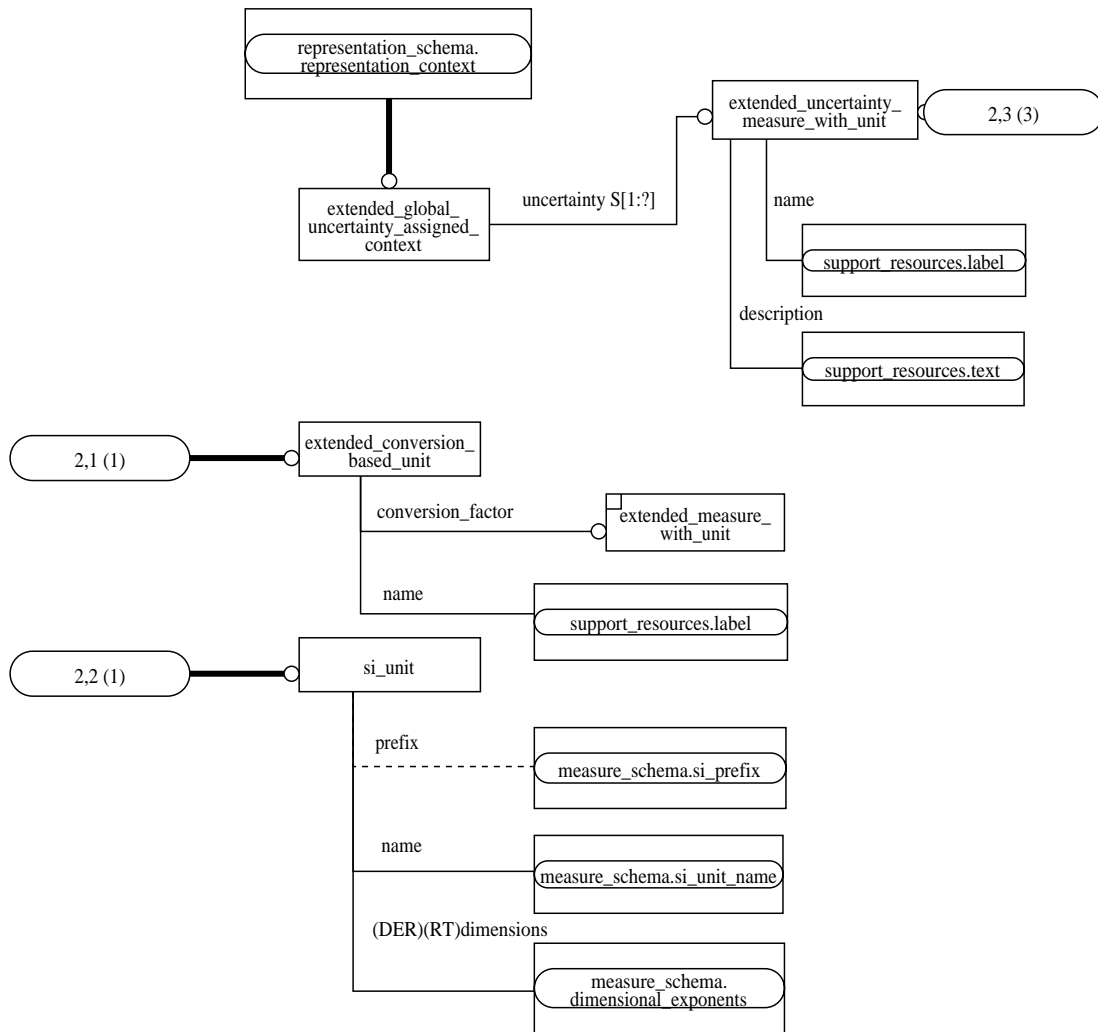


Figure G.43 – measures schema (2/4) in the ship_measures UoF

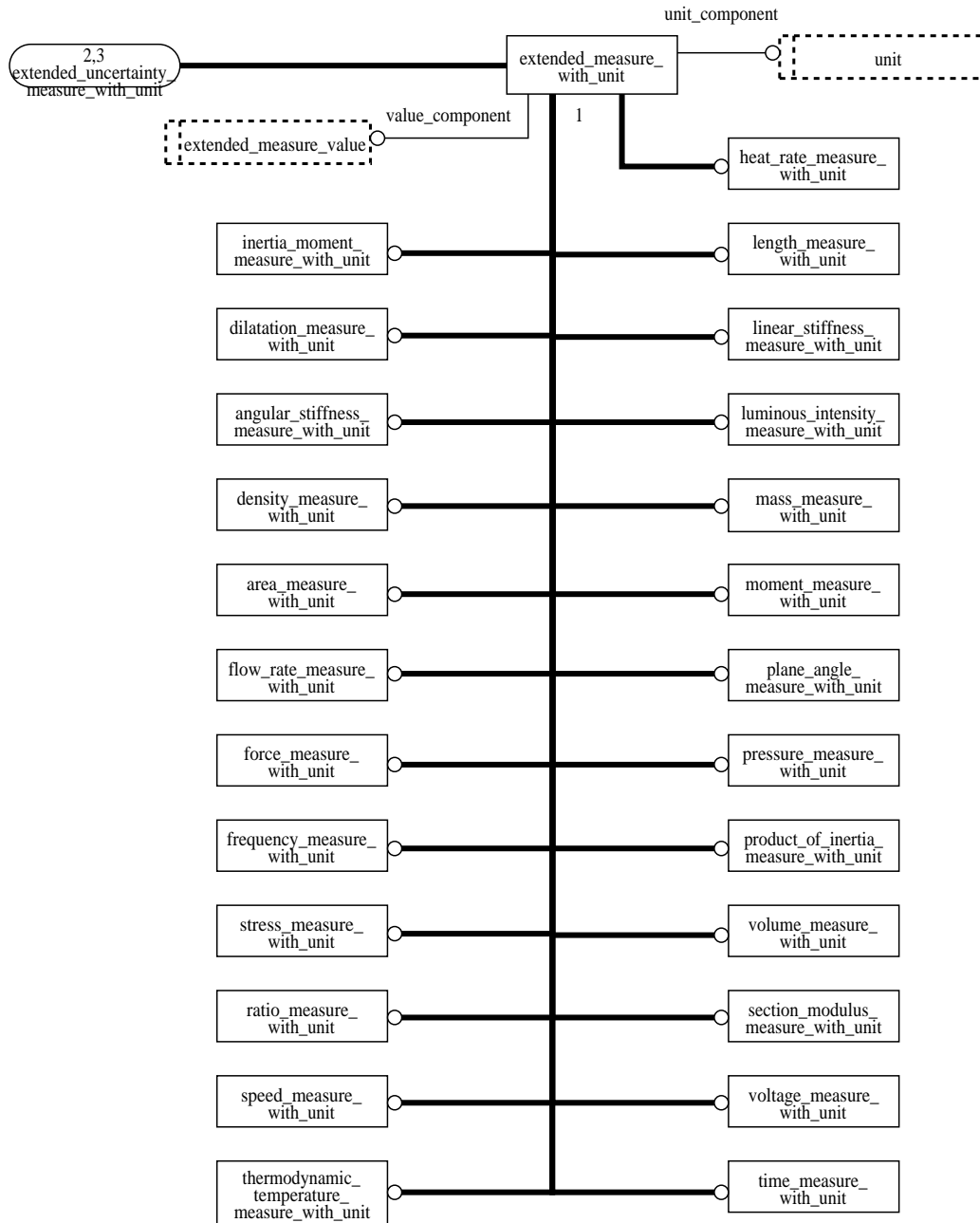


Figure G.44 – measures schema (3/4) in the ship_measures UoF

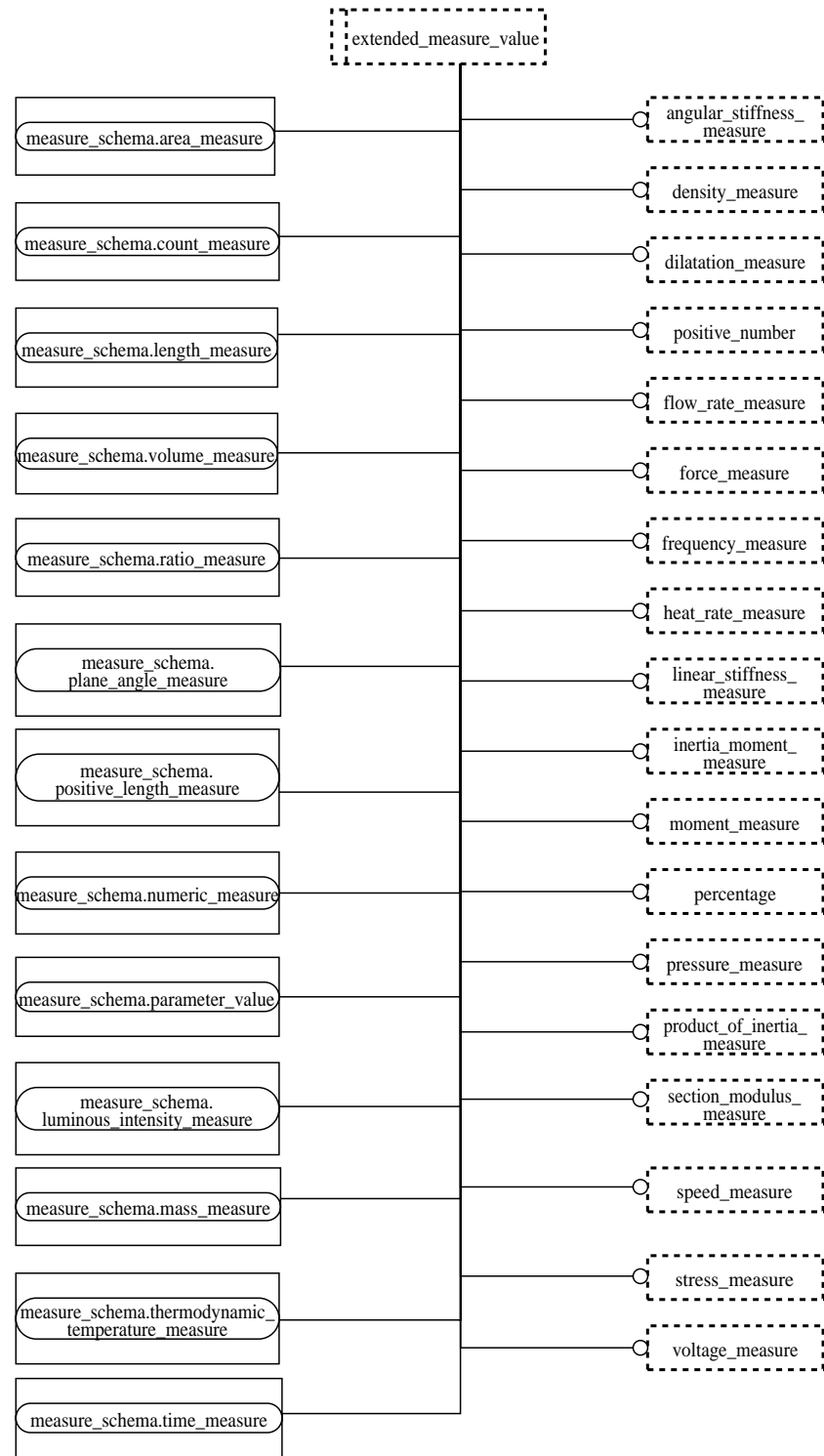


Figure G.45 – measures schema (4/4) in the ship_measures UoF

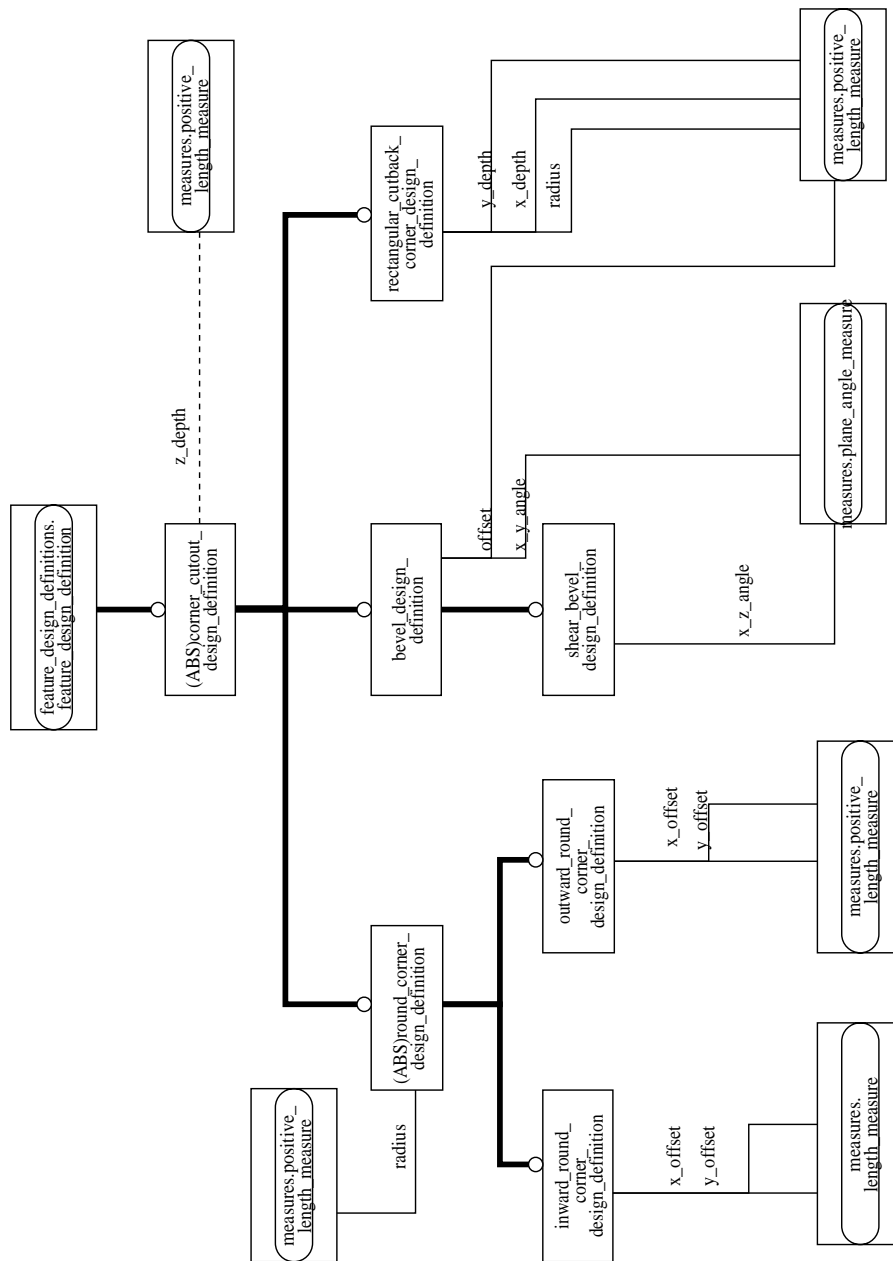


Figure G.46 – corner_cutout_design_definitions schema (1/1) in the structural_features UoF

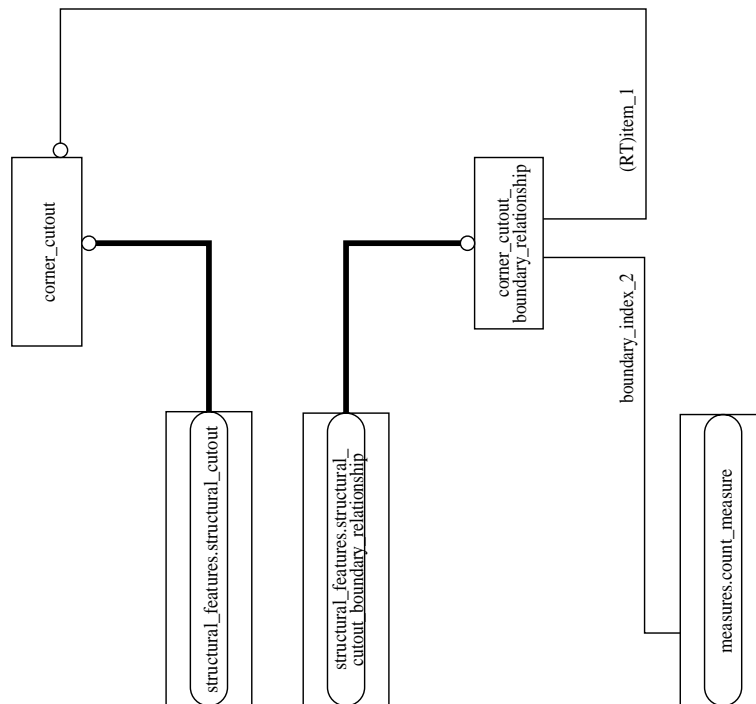


Figure G.47 – corner_cutouts schema (1/1) in the structural_features UoF

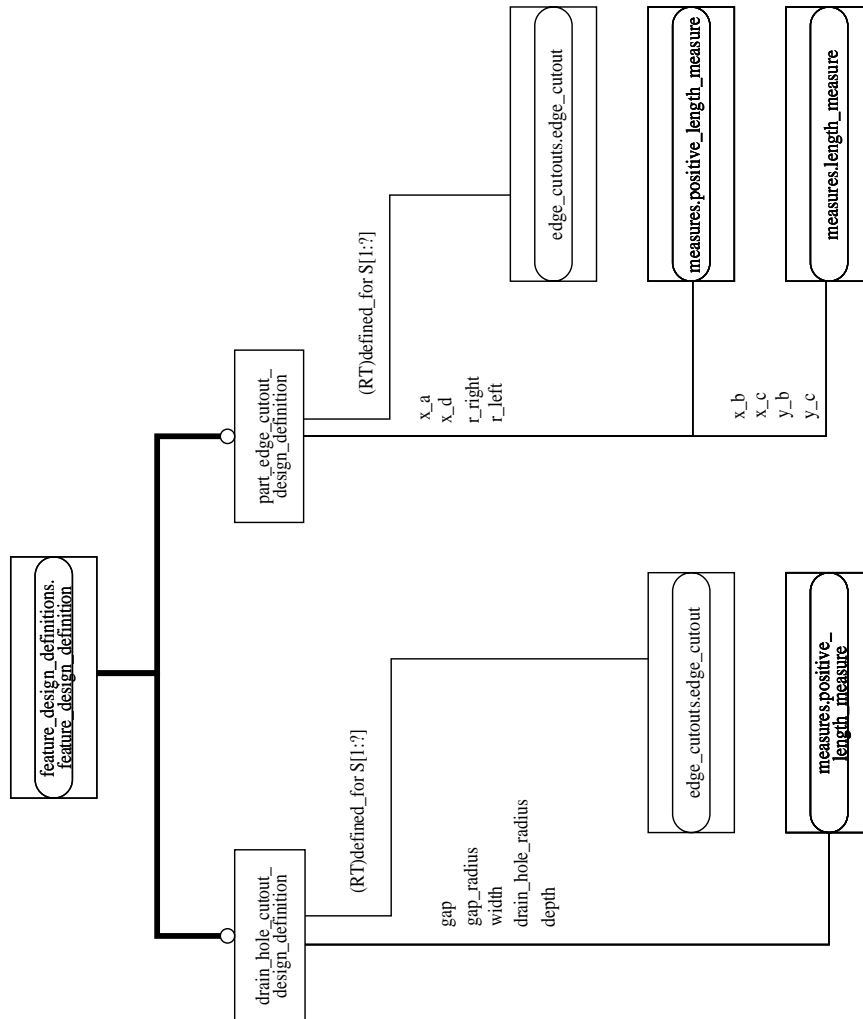


Figure G.48 – edge_cutout_design_definitions schema (1/1) in the structural_features UoF

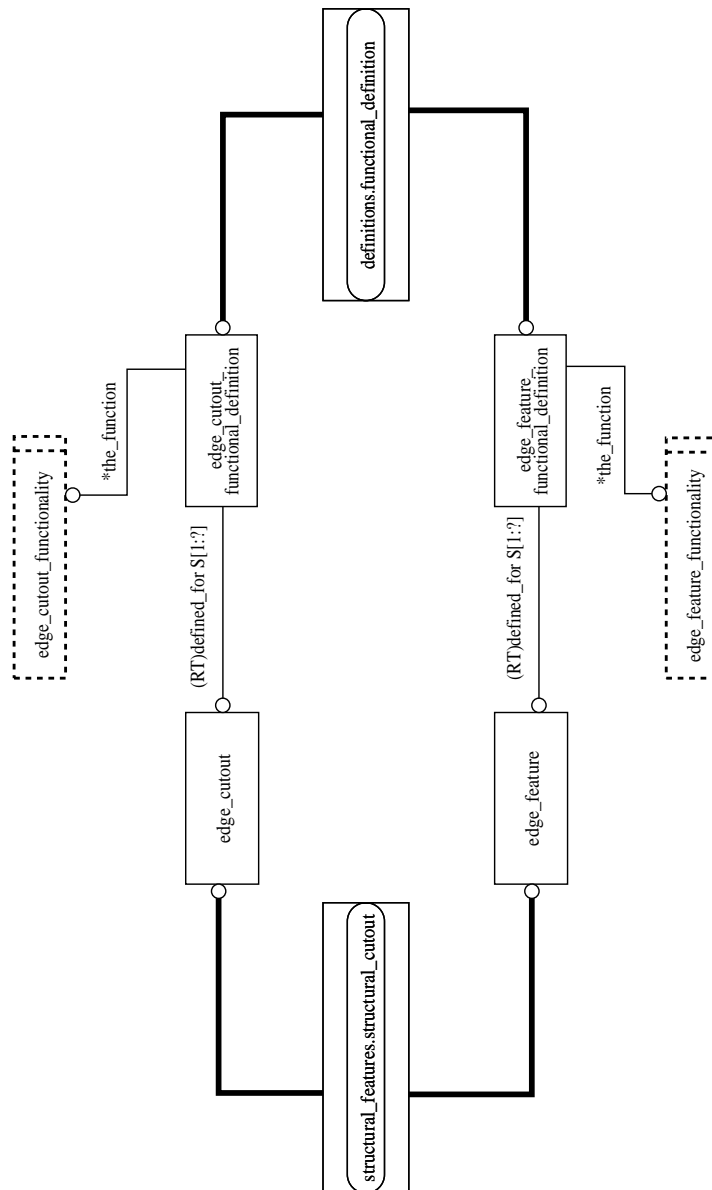


Figure G.49 – edge_cutouts schema (1/1) in the structural_features UoF

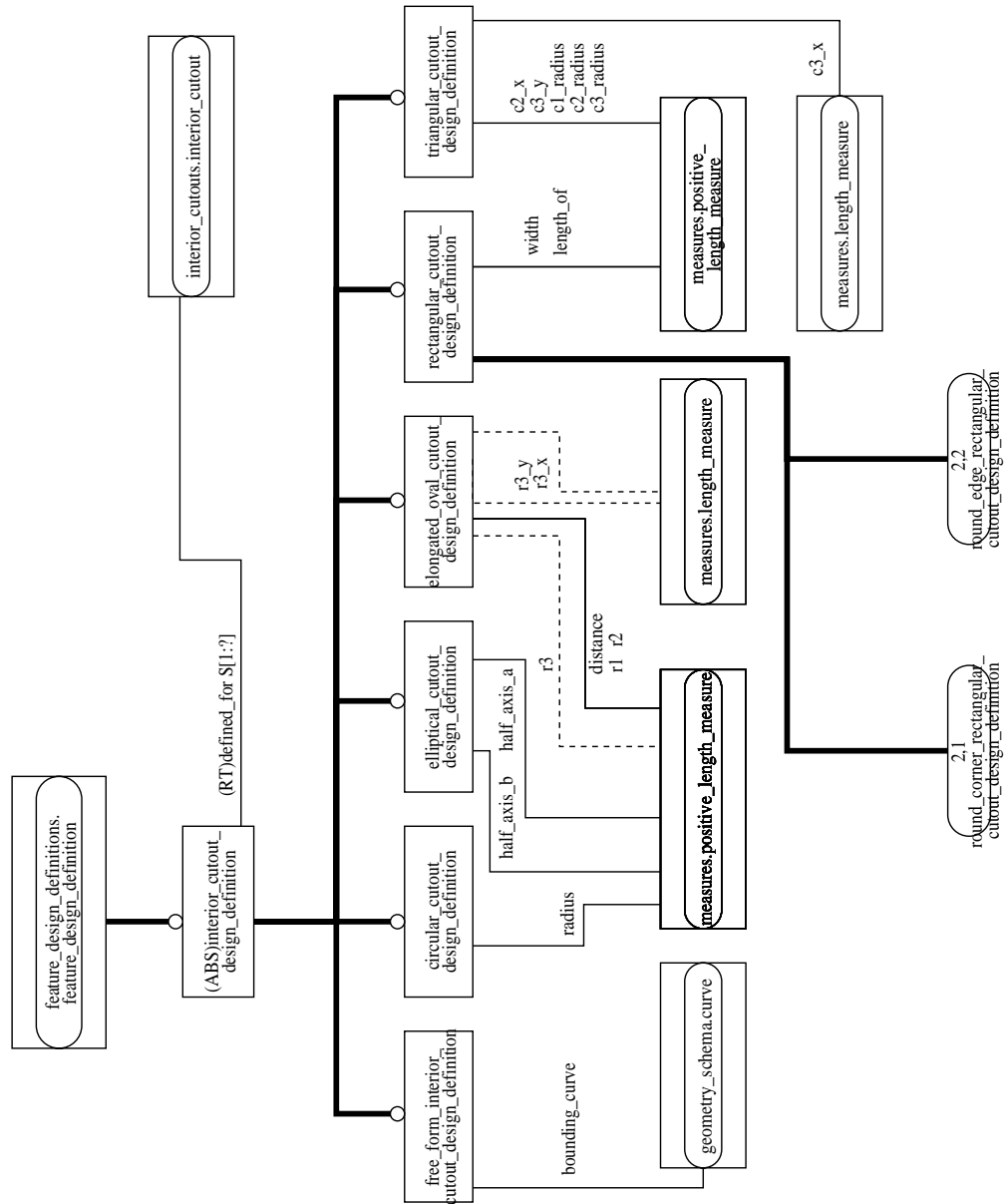


Figure G.50 – interior_cutout_design_definitions schema (1/2) in the structural_features UoF

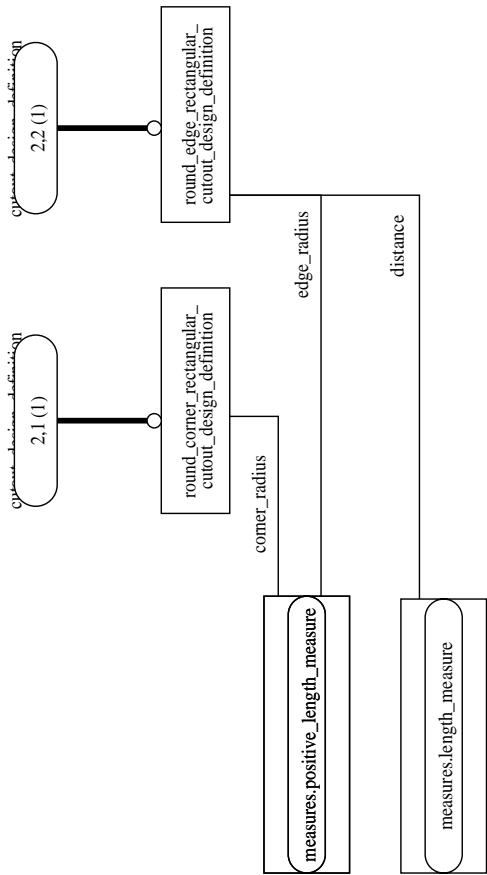


Figure G.51 – interior_cutout_design_definitions schema (2/2) in the structural_features UoF

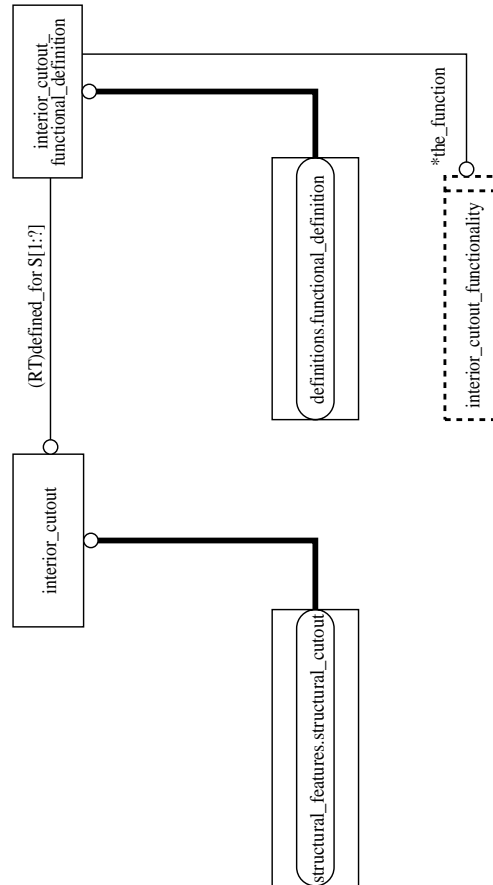


Figure G.52 – interior_cutouts schema (1/1) in the structural_features UoF

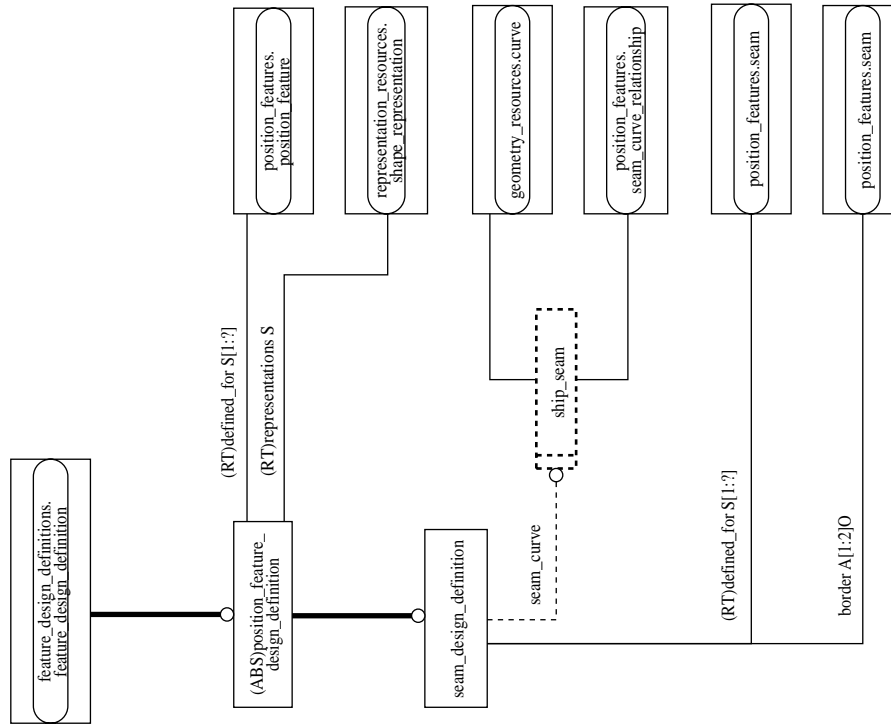


Figure G.53 – position_feature_design_definitions schema (1/1) in the structural_features UoF

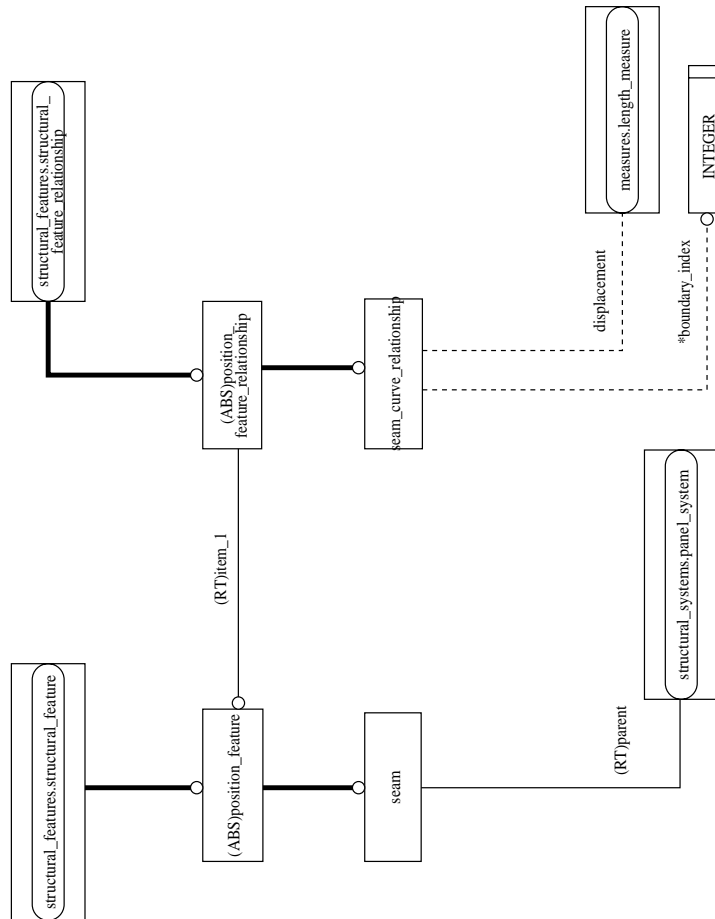


Figure G.54 – position_features schema (1/1) in the structural_features UoF

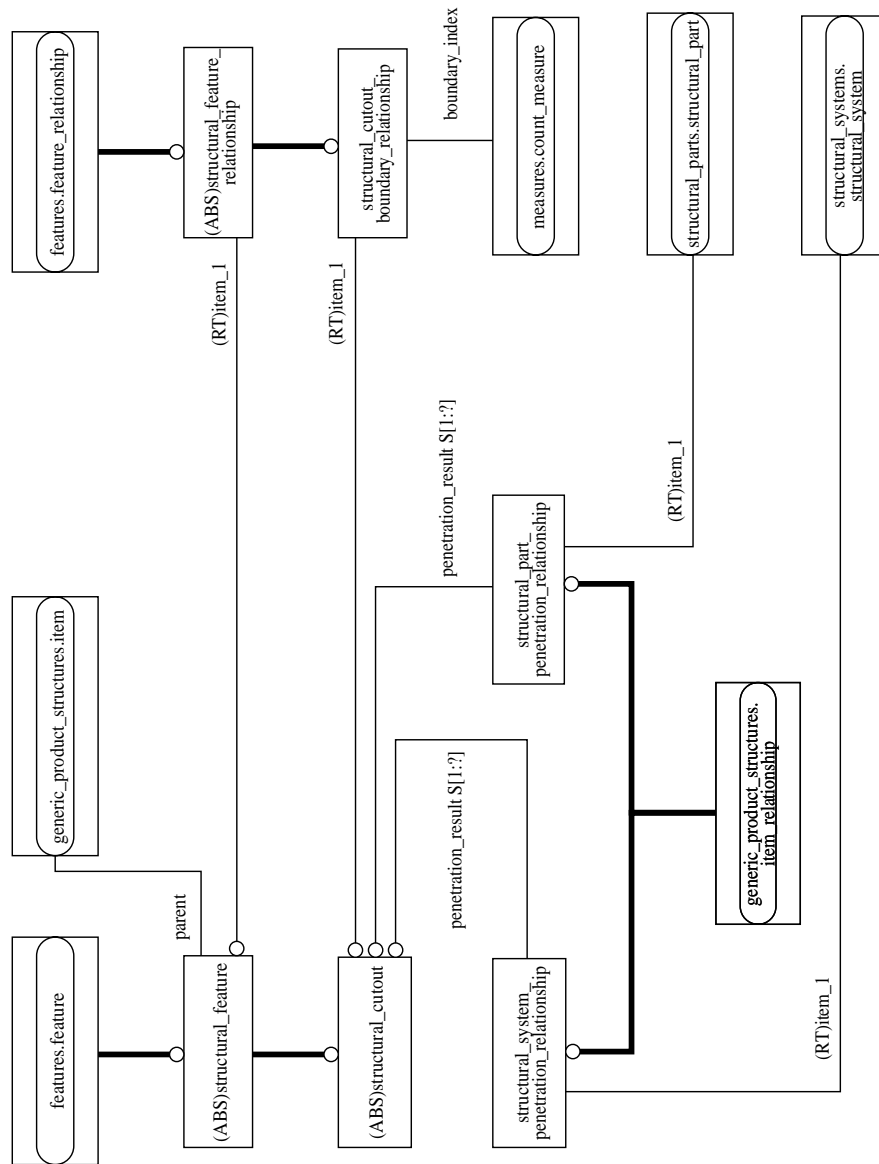


Figure G.55 – structural_features schema (1/1) in the structural_features UoF

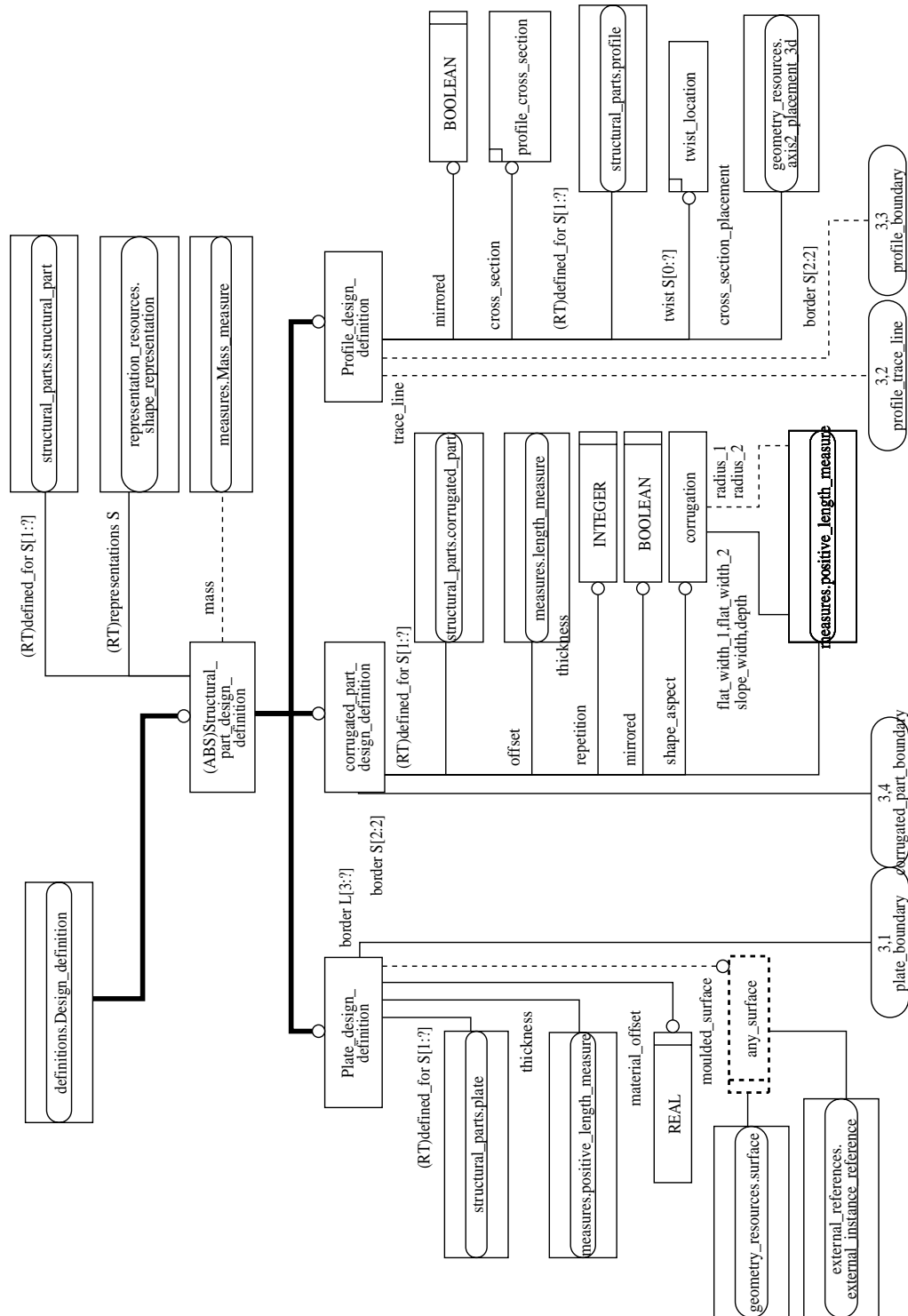


Figure G.56 – structural_part_design_definitions schema (1/4) in the structural_parts UoF

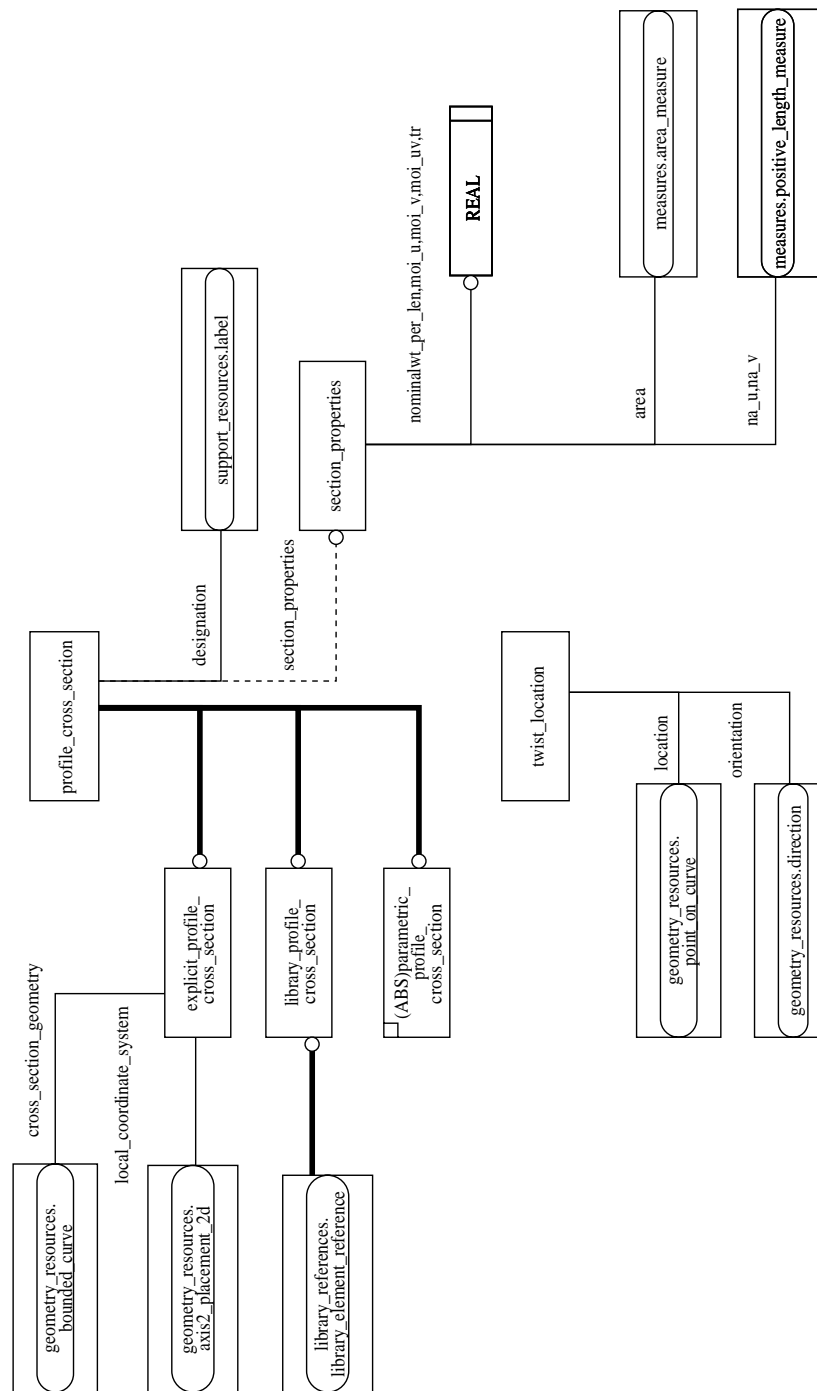


Figure G.57 – structural_part_design_definitions schema (2/4) in the structural_parts UoF

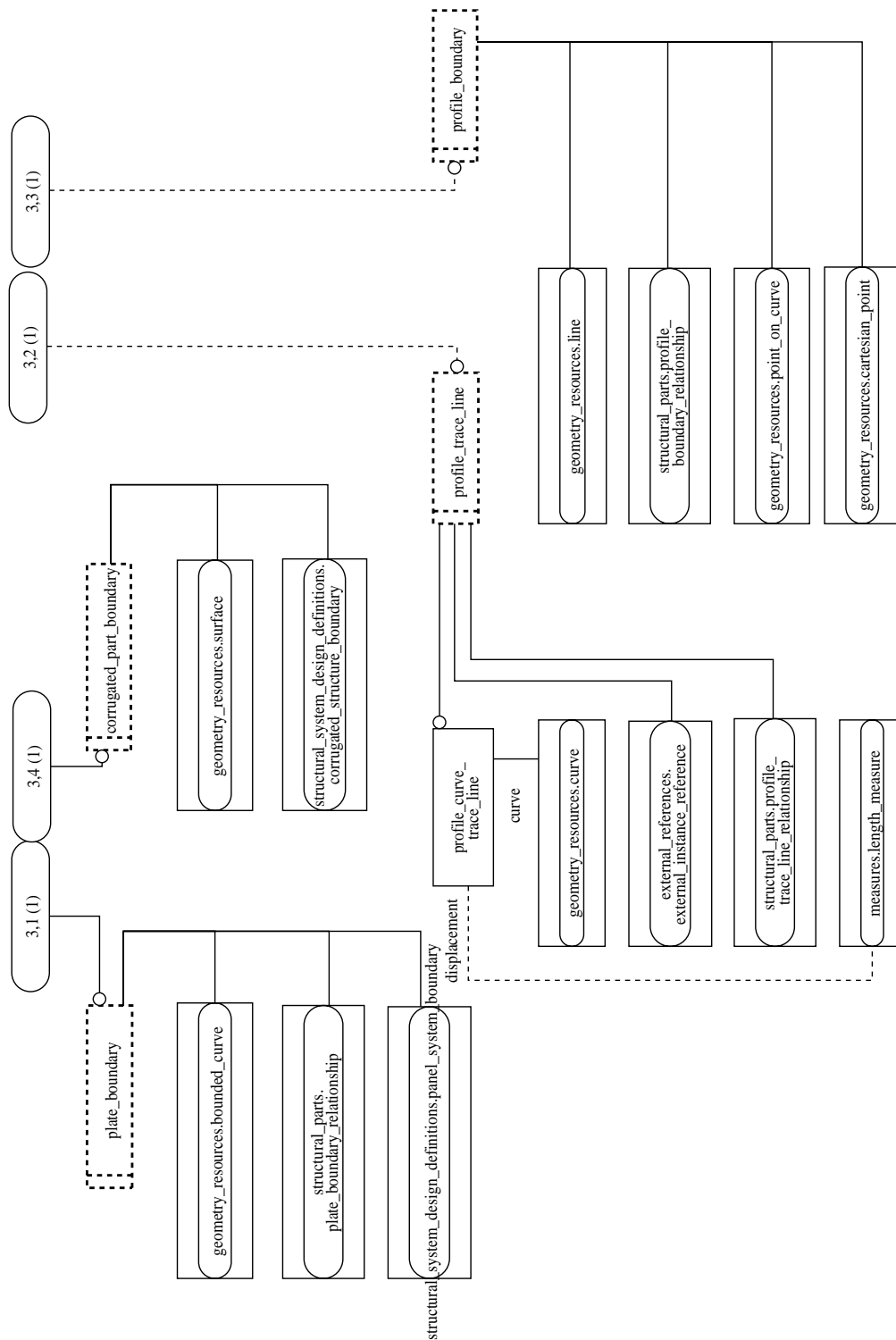


Figure G.58 – structural_part_design_definitions schema (3/4) in the structural_parts UoF

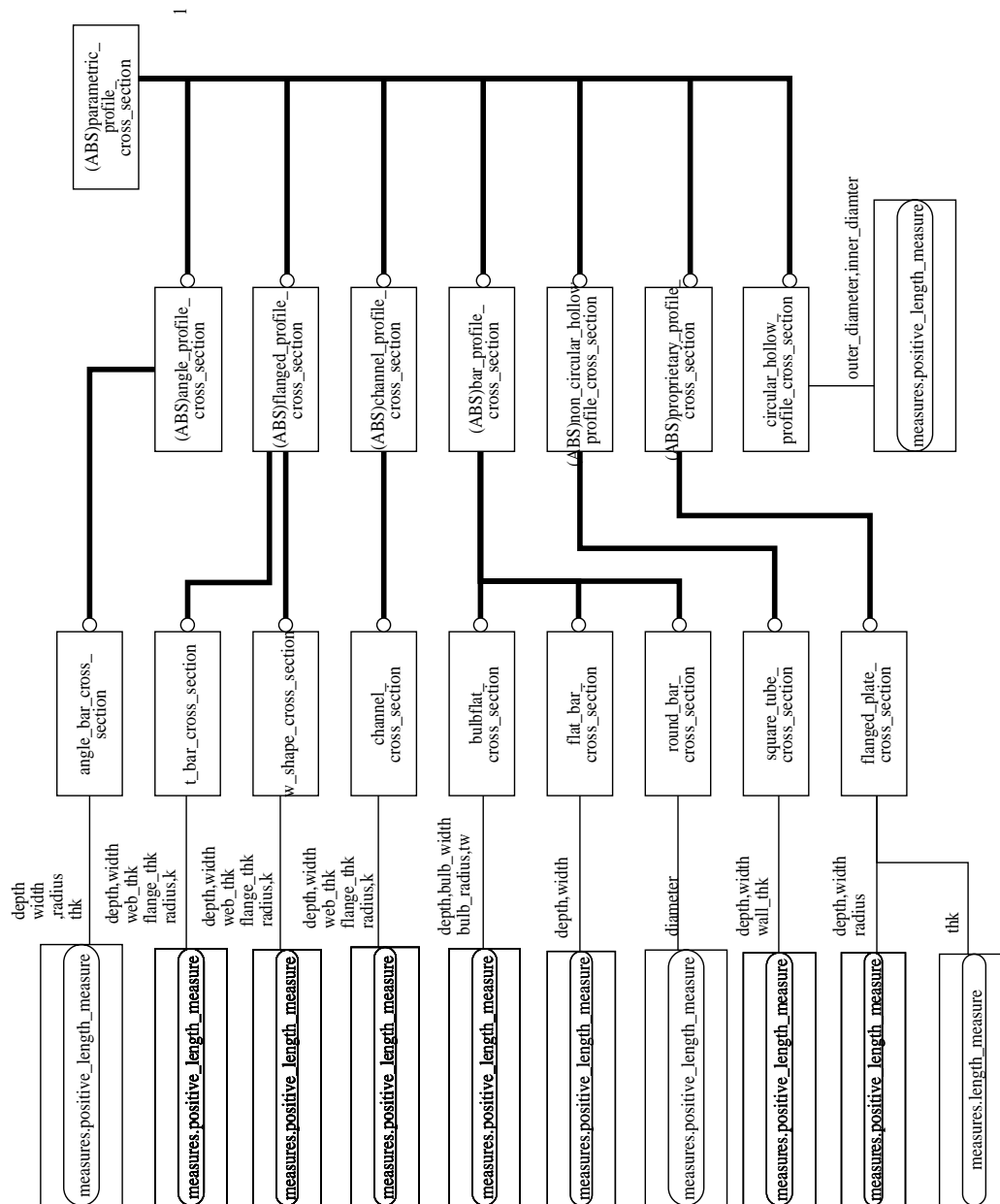


Figure G.59 – structural_part_design_definitions schema (4/4) in the structural_parts UoF

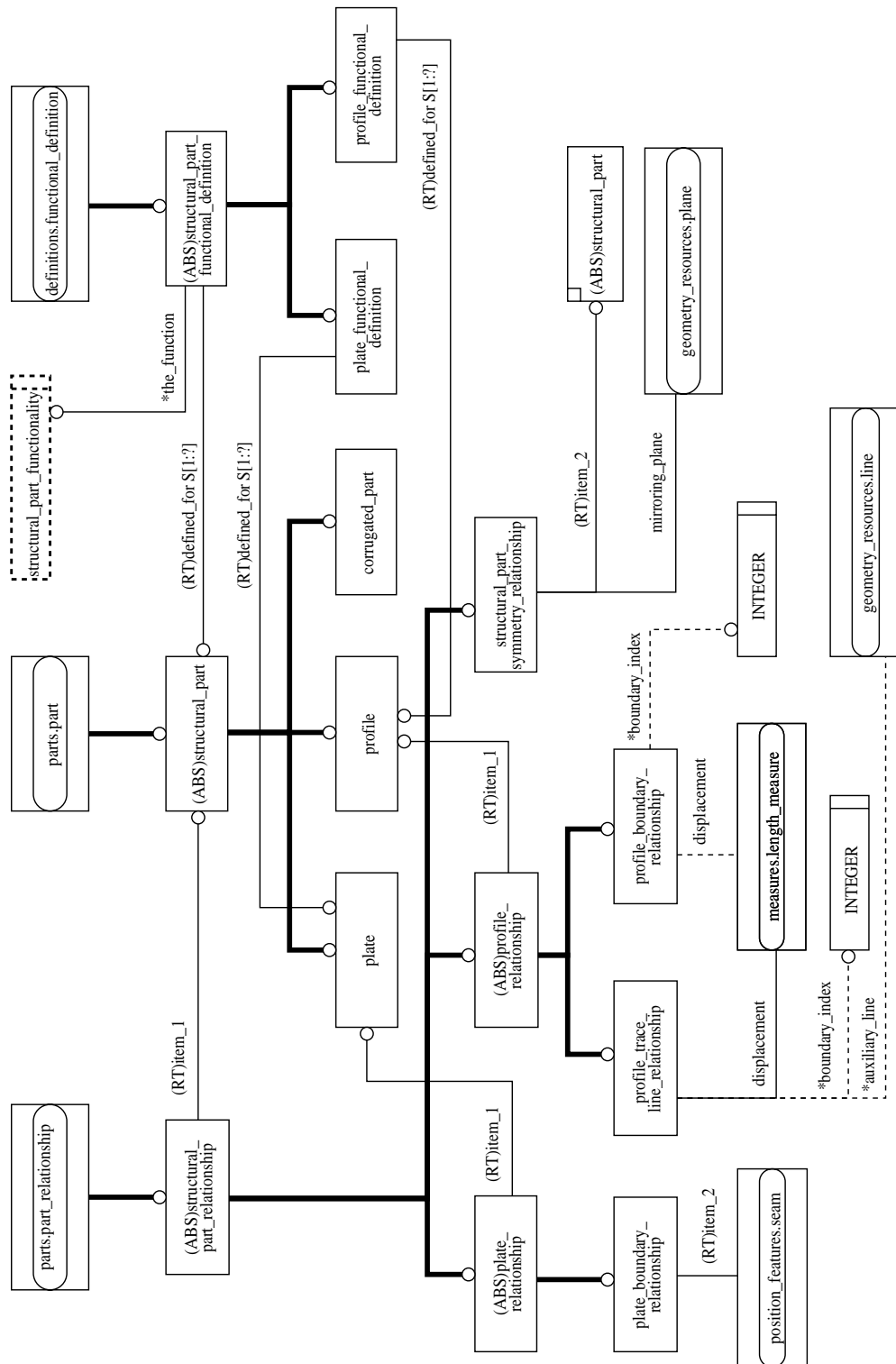


Figure G.60 – structural_parts schema (1/1) in the structural_parts UoF

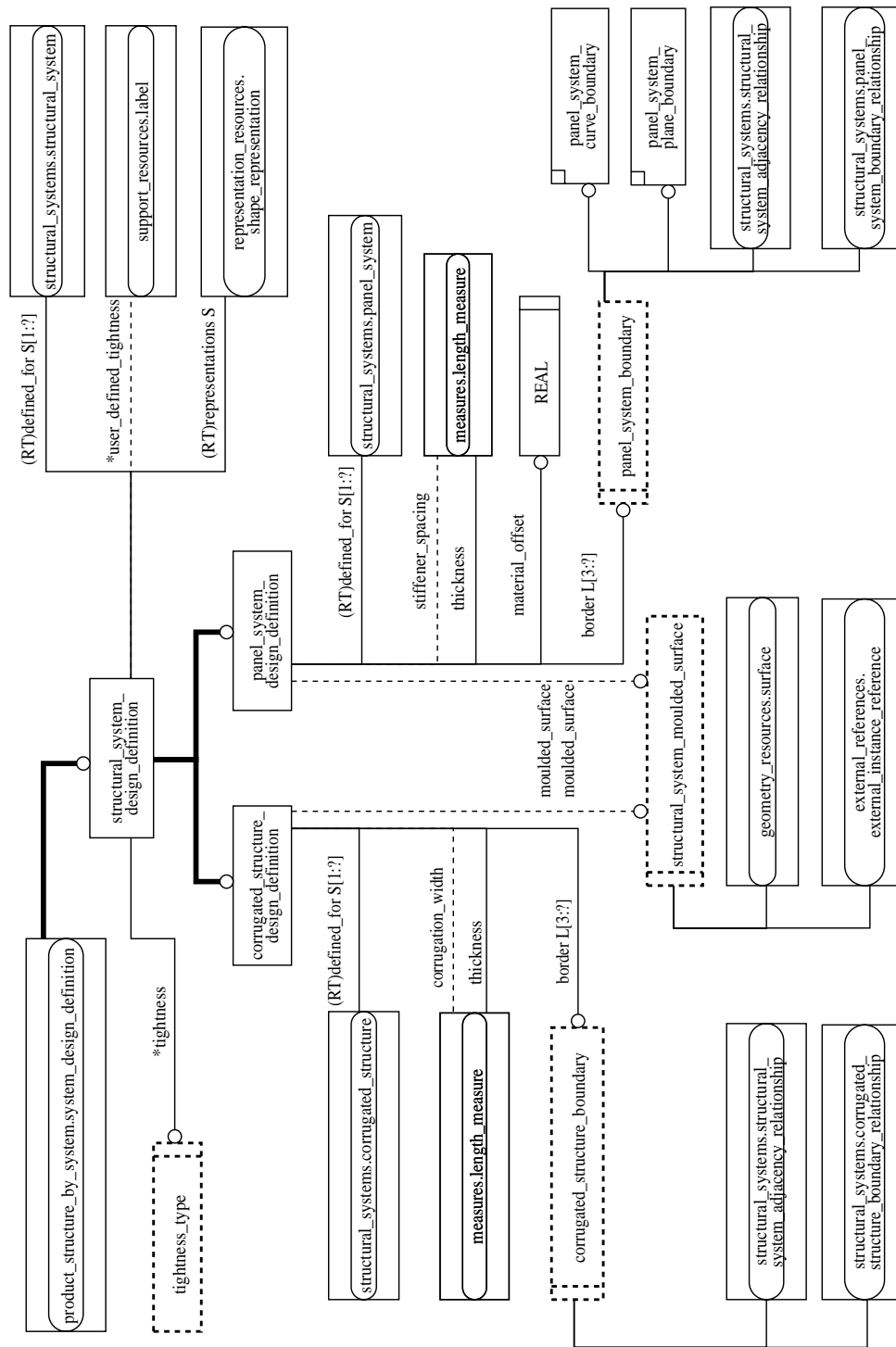


Figure G.61 – structural_system_design_definitions schema (1/2) in the structural_systems UoF

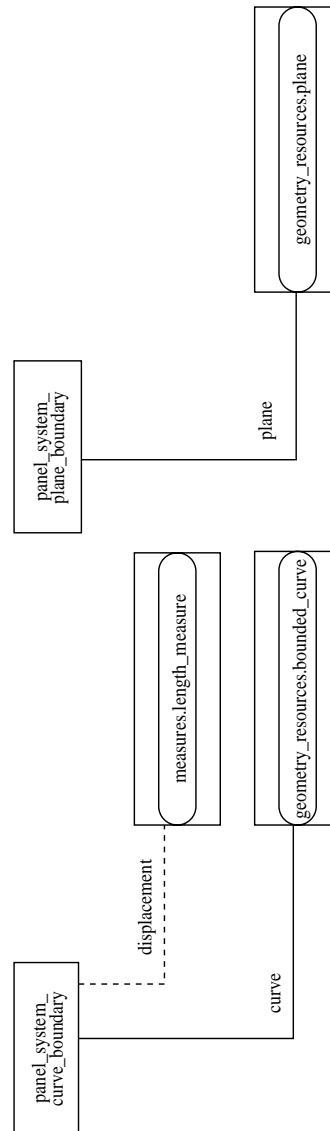


Figure G.62 – structural_system_design_definitions schema (2/2) in the structural_systems UoF

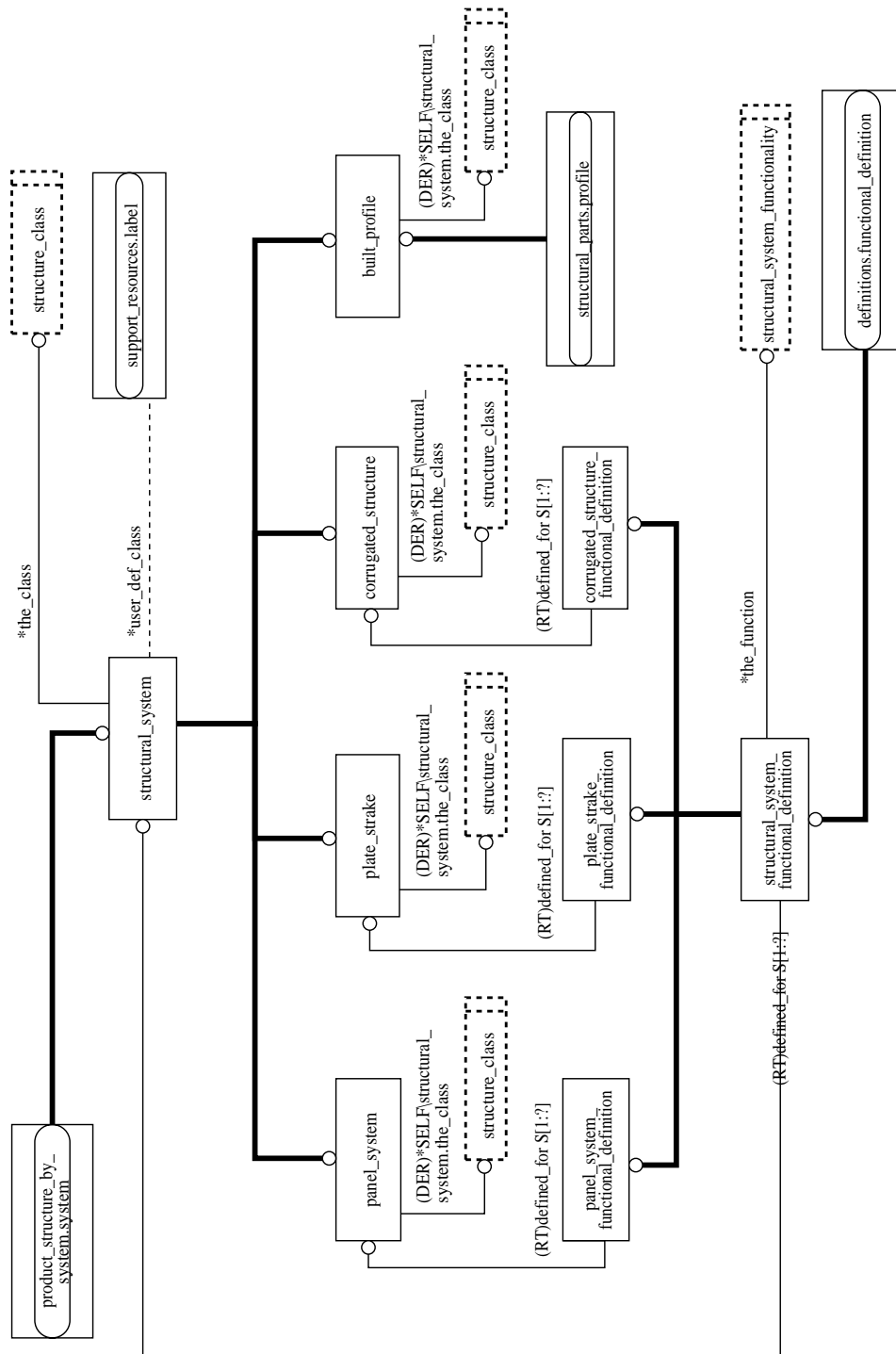


Figure G.63 – structural_systems schema (1/2) in the structural_systems UoF

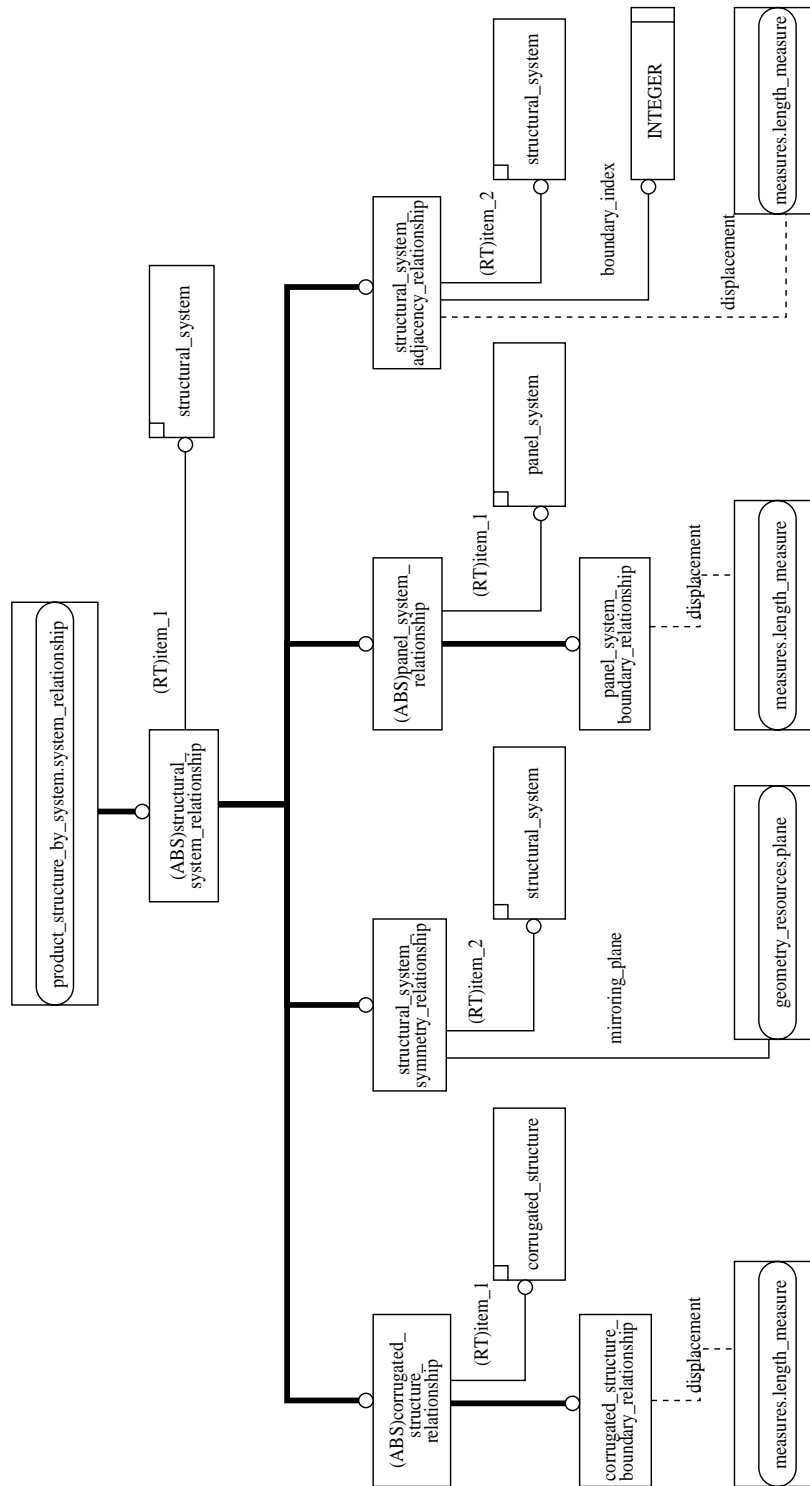


Figure G.64 – structural_systems schema (2/2) in the structural_systems UoF

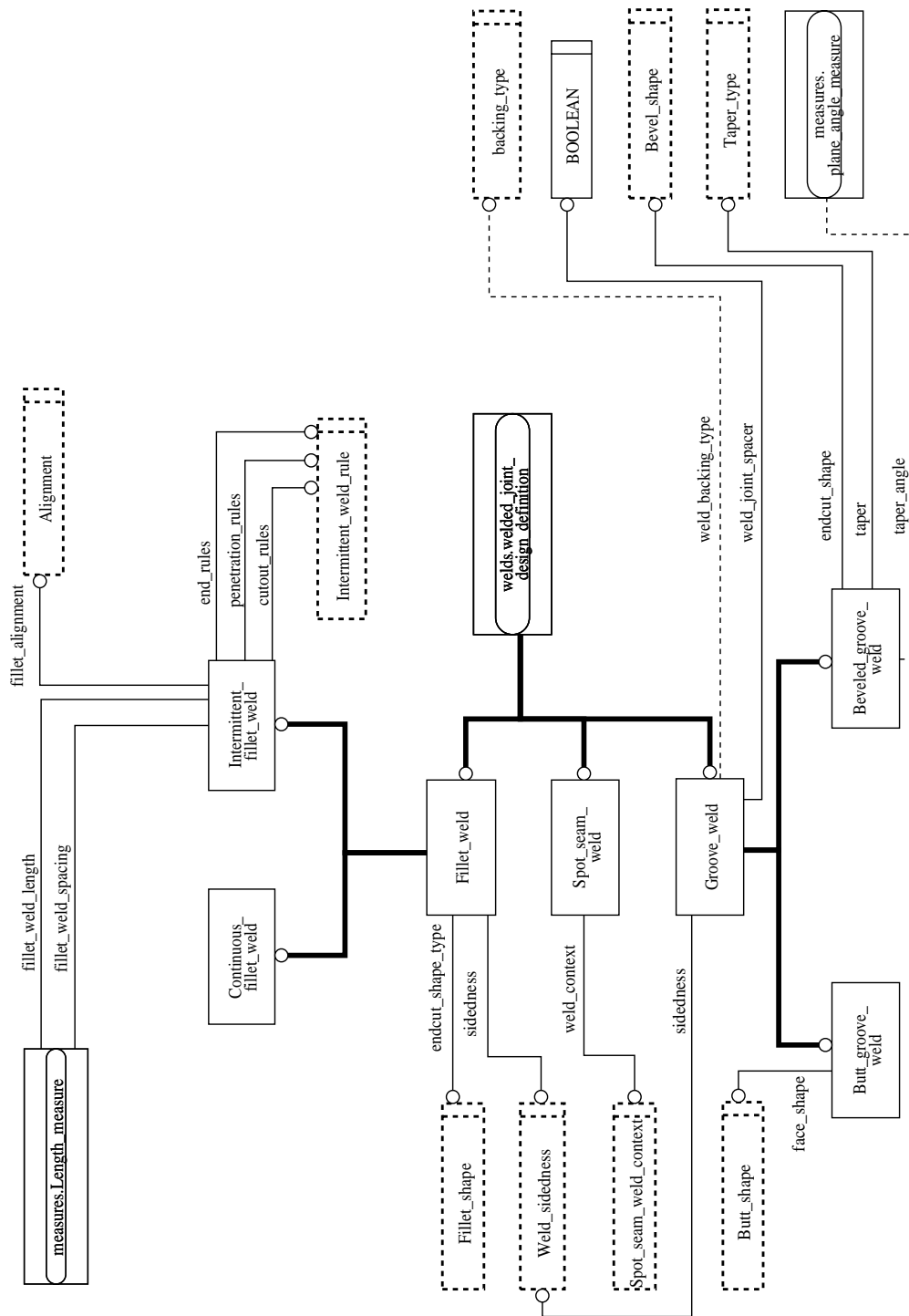


Figure G.65 – welded_joints schema (1/1) in the welds UoF

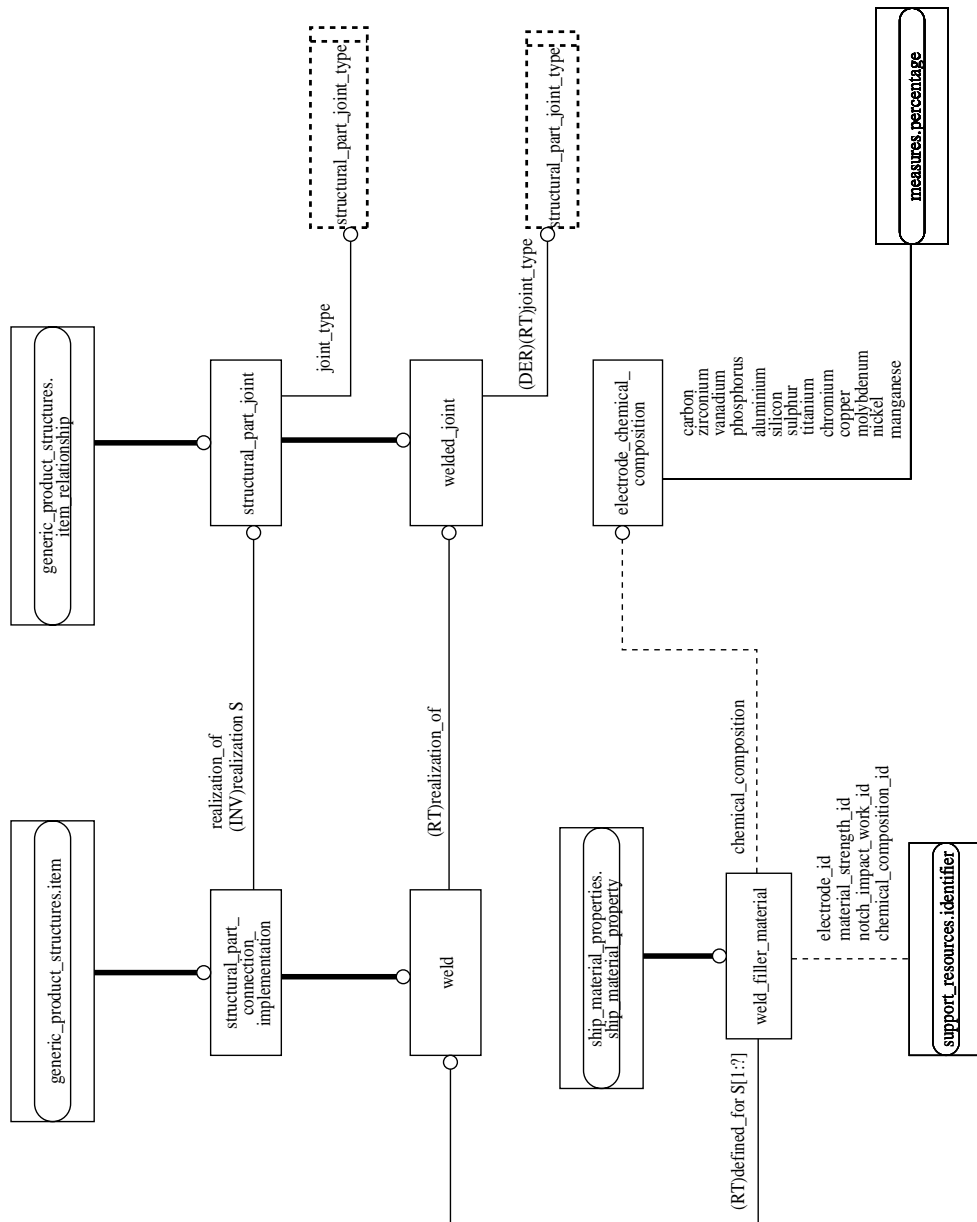


Figure G.66 – welds schema (1/2) in the welds UoF

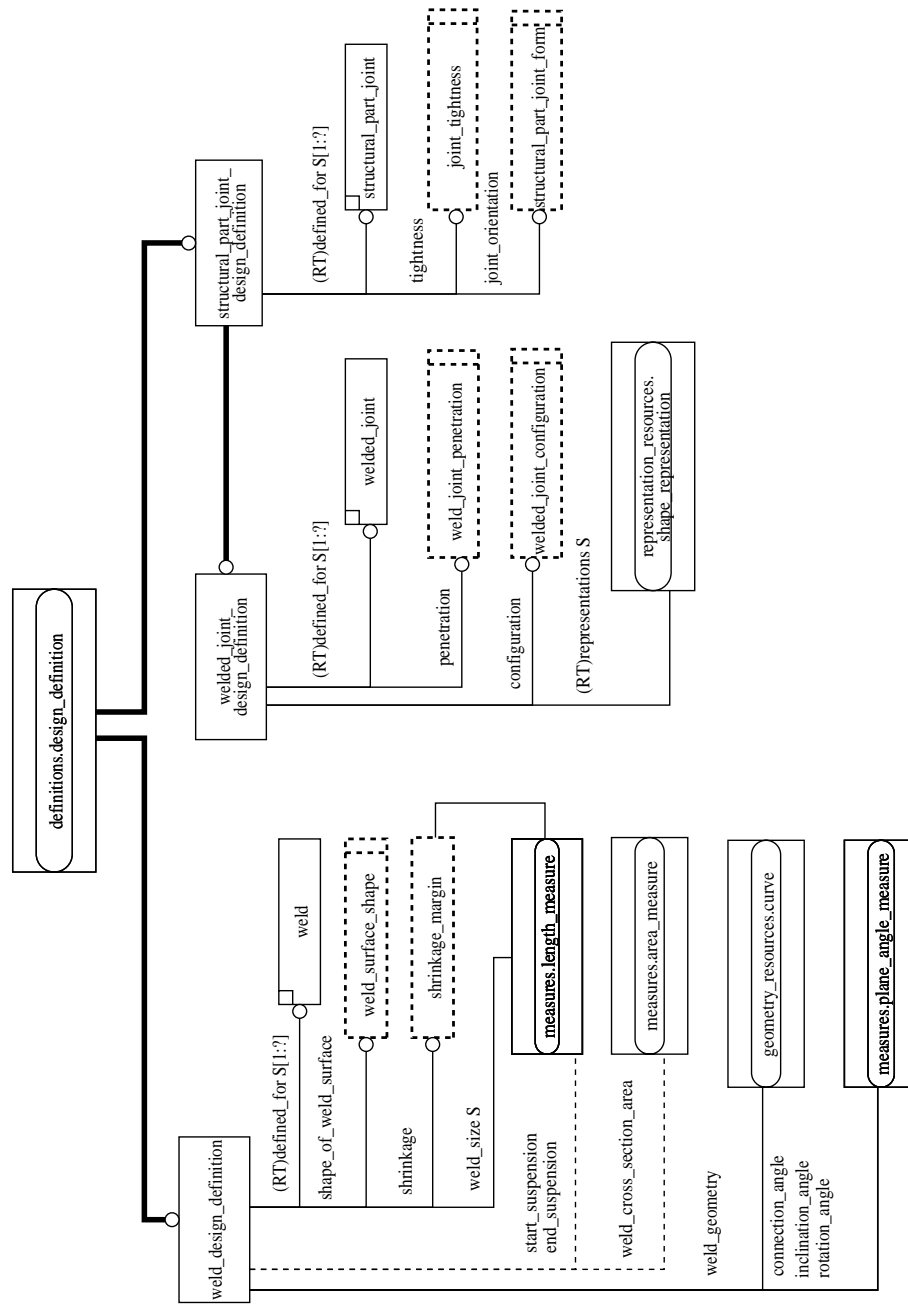


Figure G.67 – welds schema (2/2) in the welds UoF

Annex H

(informative)

AIM EXPRESS-G

Figure H.1 through H.2 correspond to the AIM EXPRESS expanded listing given in annex A. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

Annex J (informative)

AIM EXPRESS listing

This annex references a listing of the EXPRESS entity names and corresponding short names as specified in the AIM this part of ISO 10303. It also references a listing of each EXPRESS schema specified in the AIM of this part of ISO 10303, without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

Short names: <<http://www.short/>>

EXPRESS: <<http://www.express/>>

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: `sc4sec@cme.nist.gov`.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex K
(informative)

Application protocol usage guide

To be added at a later date.

Annex L (informative)

Technical discussions

L.1 Introduction

In view of the complexity of a ship it is necessary to subdivide the product model for the ship into distinct functional areas, allowing for partition into a fixed number of Application Protocols. The current version of the ship product model is shown schematically in Figure L.1.

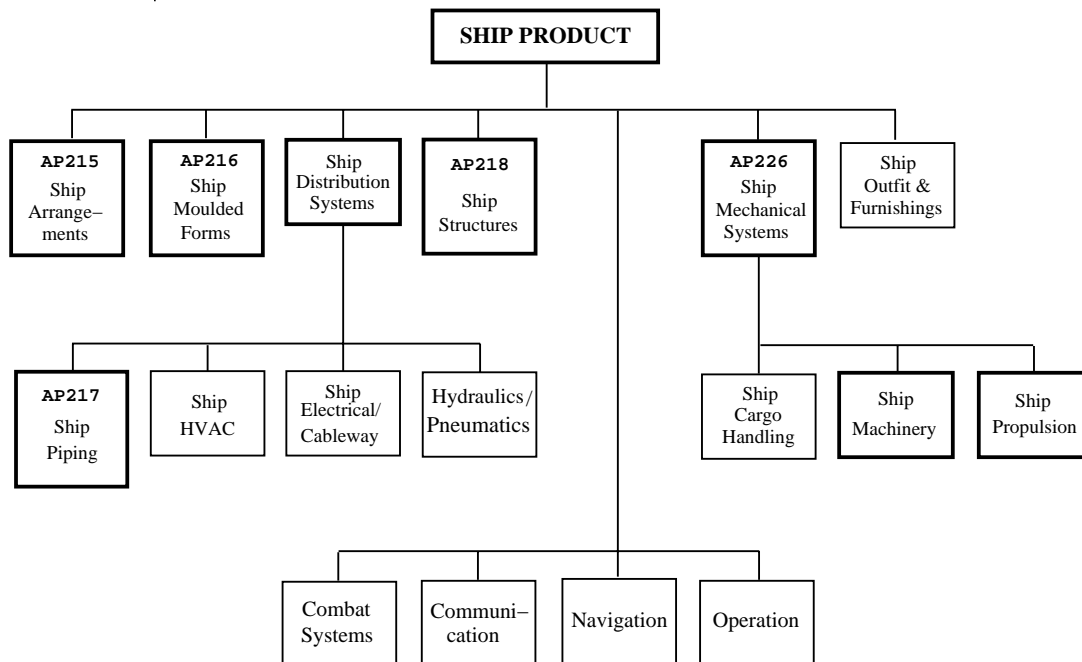


Figure L.1 – Ship product model

The key elements of the ship product model are:

- Arrangements,
- Moulded forms,
- Mechanical systems (machinery, propulsion, cargo handling),
- Structures,
- Distribution systems (piping, HVAC, electrical, hydraulics/pneumatics),

- Outfitting & furnishings,
- Communication,
- Combat systems,
- Navigation,
- Operation.

Each functional area of the ship product model is described by one or more different application protocols. The currently available shipbuilding application protocols being e under development at ISO level are:

- AP 215 Ship Arrangements;
- AP 216 Ship Moulded Forms;
- AP 217 Ship Piping Systems;
- AP 218 Ship Structures;
- AP 226 Ship Mechanical Systems.

The subdivision of the entire ship product model allows a distributed modelling work. It is also possible to start the modelling work with the functional areas reflecting the early design stages of the life cycle of the ship and validate these models before starting the modelling work at areas in the later lifecycle stages.

Each shipbuilding AP covers only a part of the ship product model. The consequence of this is that if the AP's should work together as an entire product model for the ship then an overall mechanism, around which all the shipbuilding AP's can be integrated, has to be defined. Only then the product model can be implemented in a product data management system, where all data are hold outside of application systems in the neutral data format of the ship product model. This mechanism for integrating different AP's is called the Ship Common Model, which is described in the L.2.

ISO 10303-218 has been developed by the European MARITIME and SeaSprite Projects, and supported by the European Marine STEP Association (EMSA) in cooperation with shipbuilders, designers and software developers world wide. The intent of this application protocol is to support specific application involving the ship hull structures data common in various stages of the ship life cycle, especially from the premilinary design, detailed design to the manufacturing processe of ship's structures.

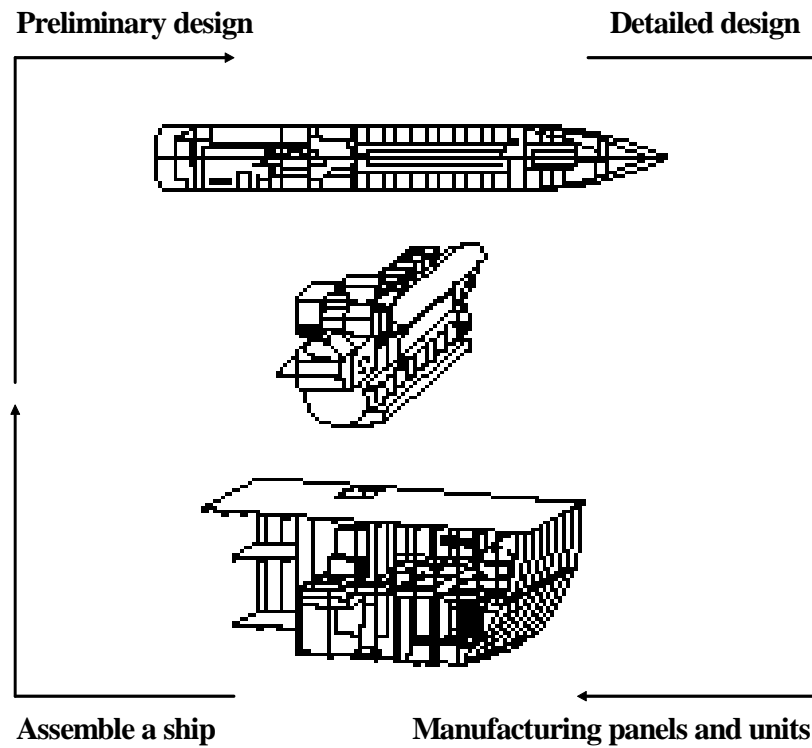


Figure L.2 – Life cycle concept of ship's structures

L.2 The Ship Common Model

The Ship Common Model defines a common framework and modelling basis for all ship-building AP's to ensure interoperability between these AP's.

The Ship Common Model (SCM) is a set of Building Blocks which are used in the ship product model context. The SCM provides a modelling framework, a set of domain (independent and re-usable) product-structure models that are required for more than one Application Protocol, as well as a set of commonly used constructs or utilities such as those used for configuration control and management concepts. The goal of the SCM is to contribute to the integration and overall consistency of the Application Reference Models of the different ship AP's.

The SCM is documented in the “AP Development Guidelines for Shipbuilding”, see Annex K Bibliography.

L.2.1 SCM framework

The modelling framework, which is part of the Ship Common Model, provides the realization of the general concepts of how to relate things, how to define their properties and how to represent them.

This framework introduces and resides in the following Building Blocks:

- definitions,
- generic_product_structures,
- representation_resources.

Effectively, this high level approach forces the product model to be split up across the main constructs of the framework namely the definable_objects, definitions and representations whilst being linked via a number of generic relationships. One of the benefits of this approach is that it enables a better management of information such as to organize the data according to different viewpoints and in the representation of life-cycle dependent requirements. Figure L.3 shows the constructure and relationship of the SCM framework elements.

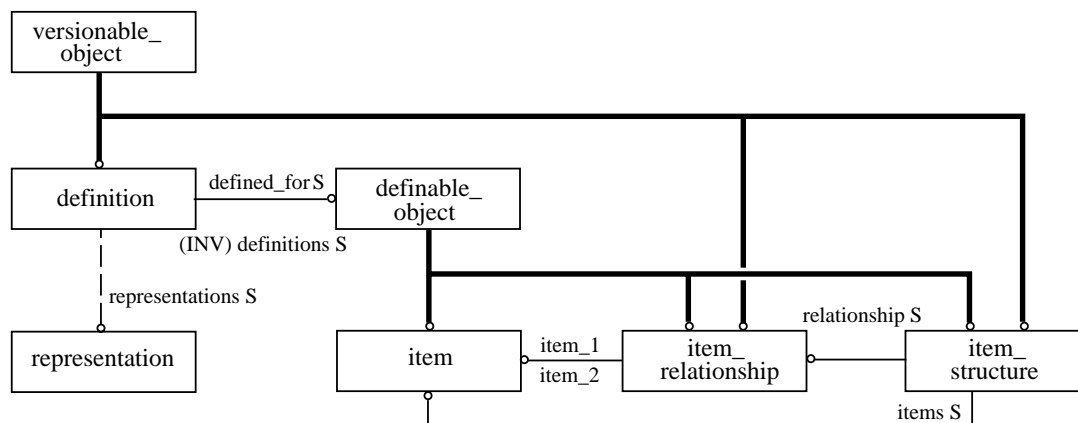


Figure L.3 – SCM framework

Fundamental to the framework is the concept of a definable_object. The definable_object is a discrete, identifiable thing used in one or more activities associated with product. It serves as the most general object from which more specialized objects are derived. A definable_object can be an item, item_relationship or item_structure. definitions describe definable_objects and are, as a result, the descriptive information-bearing entities of the

model. A definition may be further classified as a Design definition, Functional definition or Manufacturing definition, etc. A definable_object may have many different versions of definition. The definition can be changed for a definable_object during the life cycle.

Typical items in the shipbuilding AP's are

- a ship and components of the ship, such as hull, superstructure, deck, propeller, etc.;
- a part of a ship from different point of views: assembly, system, space, etc.;
- equipments, such as pump, generator, main engine, pipe, etc.;
- steel structure elements, such as plate, profile, double bottom, frame, bulkhead, etc.;
- features of steel structure elements, such holes, cutouts, endcuts, etc.;
- functional elements, such as port, logical_connections, etc.

The properties of an item are carried by the definitions. A definition must be defined for an item but an item may exist without any definition. Every property of a concept and therefore, every definition of an item, may be described in many different ways. Thus a definition can have different representations, but in some circumstances there can be a definition without representation. New representations in the shipbuilding AP's can be created by subtyping them from representation. The high level relationships between the main constructs of the modelling framework can be restricted in the subtypes of each AP. This is done by re-declarations of the attributes.

L.2.2 Domain models

The domain models provide a set of templates for organizing the product being modelled along a number of different axes or views, such as product by assembly, by system or by space. However, the templates also provide a set of implicit modelling techniques for the organization of the product data. Those domain models represent generic structures which allow the modeller to organize the data of the product according to their need. The benefit of this approach is that it reduces the modelling effort, allows consistency, conformity and interoperability with the other APs that already conform to this approach.

The intention of domain models is to create a new layer of generic elements underneath the main constructs of the modelling framework, with the possibility to restrict the attribute relations using re-declarations. These new elements introduced by the domain models are subtypes of the main constructs of the modelling framework.

The following domain models belong to UoF product_structures of the Ship Common Model:

- features,
- parts,
- product_structure_by_system,
- product_structure_by_assembly,
- product_structure_by_space,
- connection_topology.

L.2.3 Common utilities

Common utilities are a group of constructs that will be required by most shipbuilding APs. The utilities differ from the framework and domain models through the fact that the majority of cases, the utilities are ready for use and do not require any further specialization for use in an ARM. Many have been created specifically for shipbuilding although some may be able to be used externally.

Common utilities are currently available for

- ships and ship types,
- ship's general characteristics,
- configuration management,
- location concepts,
- basic geometry and topology,
- materials,
- measures and units,
- external references.

L.3 Key concepts of ship structures

L.3.1 Items and definitions of part 10303-218

ISO 10303-218 is being developed to support the exchange and sharing of ship structures data. This part has been developed in conjunction with other ship product application

protocols so that important shipbuilding concepts are common and inter-operable within the STEP parts.

The ship structures product model provides a set of subtypes of item for describing the ship structures are listed in Figure L.4. It includes ship, parts as well as their connections, systems, assemblies and hull_cross_sections as they are the essential subtypes employed in this part.

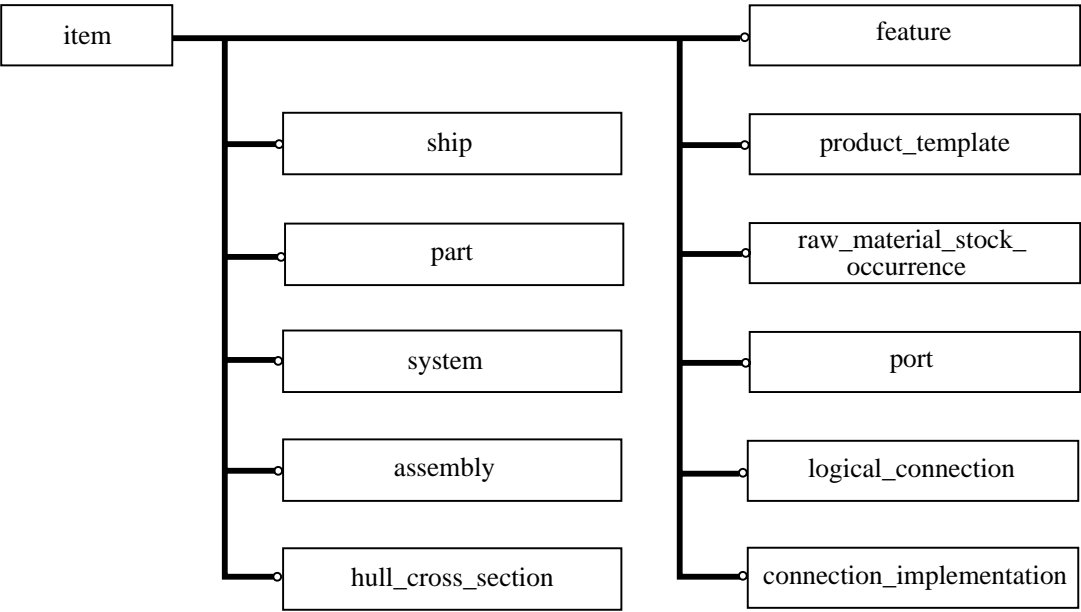


Figure L.4 – Subtypes of Items for ship structures

The subtypes of Definitions for this part are listed in Figure L.5. This set of the definition sybtypes serves to describe the design and manufacturing data of ship structures, including the necessary locational and functional information.

The concept of item in this part is used via a hierarchical partitioning. The nature of such a

The Definition for the structures may be classified as a Design_definition, which is a main definition part for the preliminary and detailed design of the ship’s lifecycle stages. The main components of the Design_definition types, such as design definitions for structural parts, structural part features, structural part joints and structural systems, etc. employed in this part of ISO 10303 are shown in Figure L.6.

L.3.2 Structural systems

Structural_system is a specific System of this part 10303-218. It provides information and capabilities common to all types of ¡Structural_system¿s. The ¡Item¿s carried by a

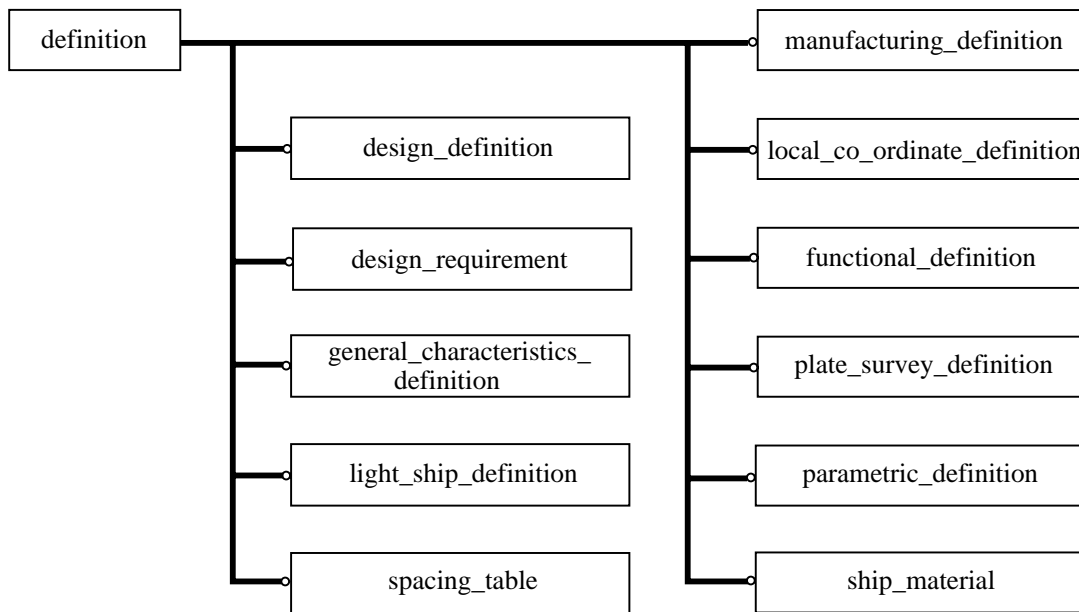


Figure L.5 – Subtypes of Definition for ship structures

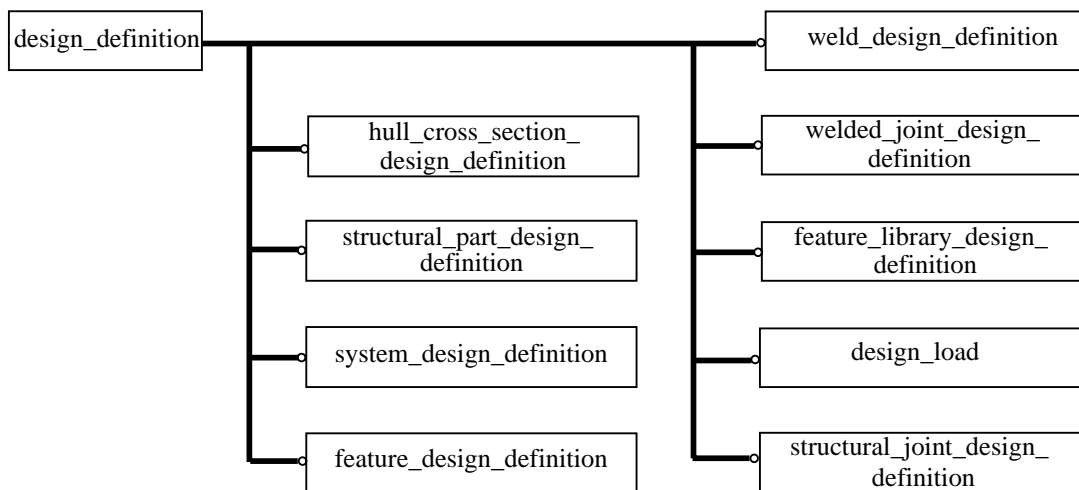


Figure L.6 – Subtypes of Design_definition for ship structures

¡Structural_system¿ are restricted to be of type ¡Structural_system¿ or ¡Structural_part¿. The same holds for the relationships.

A ¡Structural_system¿ is of a specific class, as specified by the class attribute:

SECTION: an aggregate of other (sub)sections, blocks and/or ¡Structural_part¿s; has both a strength ****AND**** a tightness function

PANEL_SYSTEM: special section that consists of one to many plates reinforced by profiles running in the same main direction; the plates lie in the same (smooth mathematical) moulded surface (plane or non-plane)

CORRUGATED_STRUCTURE: - special section that consists of only `Corrugated_part`s and `Plate`s

BLOCK: also referred to as constructional group; an aggregate of other (sub)blocks and/or a tightness function

PLATE_STRAKE: special block that consists of only plates; has a tightness function

BUILT_PROFILE: a `Structural_system` that is made of at least two `Structural_part`s (e.g. `Plate`s or `Profile`s) by an aggregation process; can nevertheless be used like a profile

L.3.3 Corrugated parts and systems

Corrugated structures required for this AP218 are:

- the structures have not only rectangular overall shape, e.g., tank tops may have rises of a floor, bulkheads may have waved plates;
- corrugations run along one axis, linearly extruded along the other;
- shape expressed using a set of parameters;
- the corrugations support for assembled purchased structures and those purchased section-wise and welded together;
- they are bounded or trimmed by other structural elements.

Figure L.7 is EXPRESS diagram of the corrugated—`part` model.

Figure L.8 is EXPRESS diagram of the corrugated—`system` model.

L.3.4 Structural features

`Structural_features` specifies the concepts for the definition of Features that are applied to `Structural_parts` and `Structural_systems`.

The shape of a `Structural_feature` may be represented in one of the following ways:

- (1) by explicitly describing the shape using `Explicit_feature_design_definition` (BB `feature_design_definitions`);

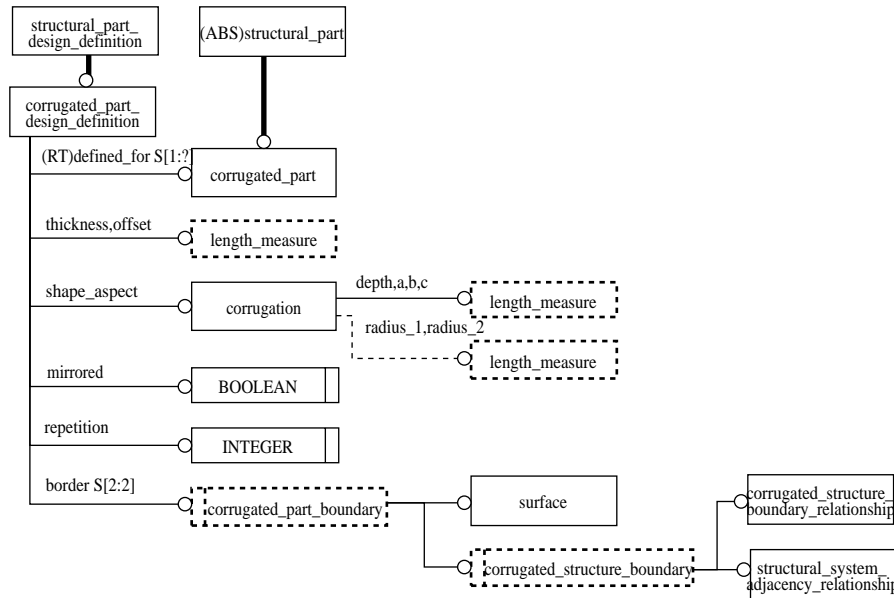


Figure L.7 – Corrugated part model

- (2) by referring to an entry in a features library using Feature_library_definition (BB feature_design_definitions);
- (3) by describing the shape parameters using one of the feature type specific Design_definitions; these are:
 - Free_form_interior_cutout_design_definition,
 - Circular_cutout_design_definition,
 - Elliptical_cutout_design_definition,
 - Elongated_oval_cutout_design_definition,
 - Rectangular_cutout_design_definition and its subtypes,
 - Triangular_cutout_design_definition and its subtypes,
 - Part_cutout_design_definition,
 - Drain_hole_cutout_design_definition,

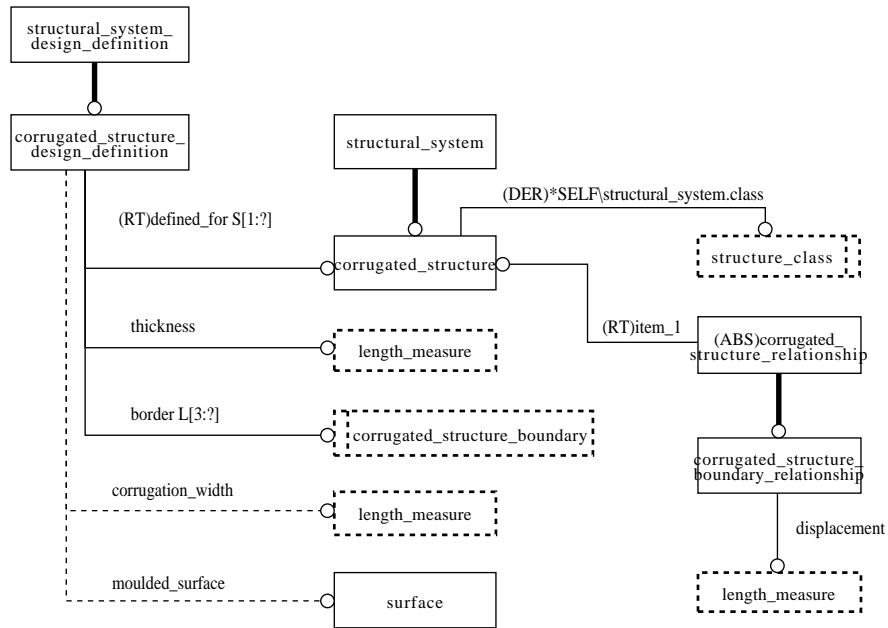


Figure L.8 – Corrugated system model

- Round_corner_design_definition,
- Inward_round_corner_design_definition,
- Outward_round_corner_design_definition,
- Bevelled_corner_design_definition,
- Shear_bevelled_corner_design_definition,
- Rectangular_cutback_corner_design_definition.

No reference to moulded geometry is provided, except via the parent of the Feature if this link exists (via the corresponding relationship-entities) and if the parent references underlying geometry.

A hierarchy of feature design definitions can be shown in Figure L.9.

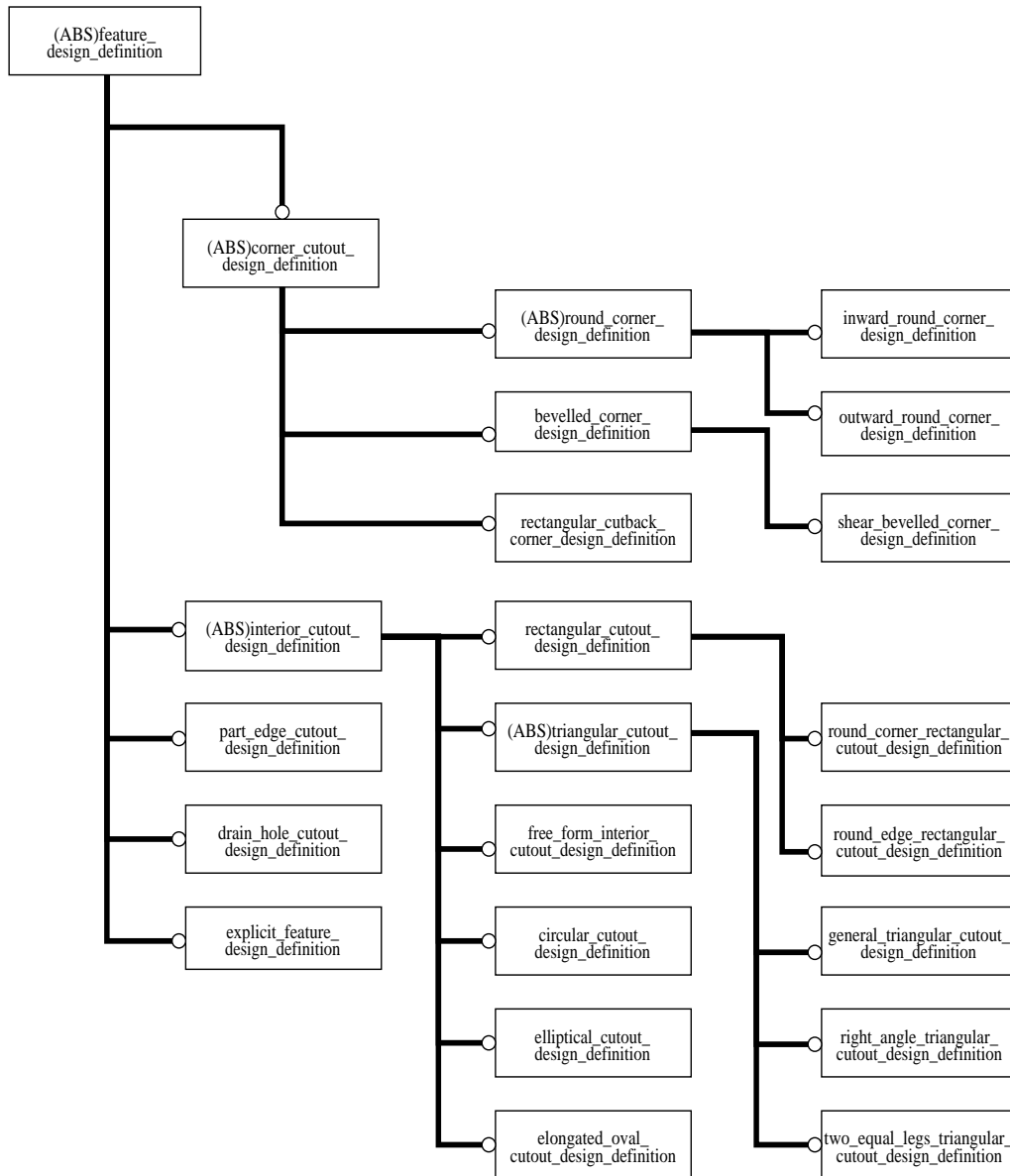


Figure L.9 – Hierarchy of feature design definitions

Bevelled_corner_cutout

A bevelled_corner_cutout (or chamfer), ISO/DIS 10303-224 denotes the surface obtained when intersecting a rectangular work item with a plane having a normal vector not parallel to any of the adjacent surfaces of the work item. The origin of the local co-ordinate system is placed at the intersection of the work item boundary lines at the corner that is to be cut-off. The axis of the local co-ordinate system coincides with the work item edges and point such that the cut-off occurs always in positive axis direction and such that offset and the angles are measured along and from the x-axis.

- Bevelled_corner_cutout: The normal vector of the bevel surface is perpendicular to the normal vector of one adjacent work item surface.
- Shear_bevelled_corner_cutout: The normal vector of the bevel surface is not perpendicular to any of the normal vectors of adjacent work item surfaces.

Circular_cutout

A circular_cutout is an circular opening in a work item specified by a radius "r". The mathematical description is given by: $x^2 + y^2 = r^2$. The local origin coincides with the centre of the circle.

Elliptical_cutout

An elliptical_cutout is an opening in a work item described by an ellipse. The mathematical description of the ellipse is given by: $(x/a)^2 + (y/b)^2 = 1$. The local origin is placed at the centre of the circle and the co-ordinate axis (x-axis and y-axis) coincides with the ellipses half axis as shown below. "a" and "b" are the half axis parameter.

Elongated_oval_cutout

An elongated_oval_cutout is an opening shaped like an egg or a slot. The basic form consist of two circles, "r1" and "r2", separated by a "distance" and joined by an envelope. The default envelope consist of two straight lines joining the two circles. If the two circles have the same radius the opening is a slot. By specifying an additional circle arch (i.e. radius "r3" and arch centre co-ordinates "r3_x" and "r3_y") the opening obtains an egg shaped form. The local origin is placed at the centre of the left circle. The local x-axis is given by the line through the centres of the two basic circles (= symmetry axis).

Rectangular_cutout

A rectangular_cutout is an opening where the opposing sides of the opening are parallel and adjacent sides are perpendicular. The interior angles of the opening are 90 degree. A "width" and "length_of" specify the rectangular opening. The local origin is placed at the intersection of the symmetry axis of the opening.

Round_corner_rectangular_cutout

A `round_corner_rectangular_cutout` is an `rectangular_opening` where the corners of the opening are rounded off. The rectangular opening is specified by a "width", "length_of" and the corner radius "r". The local origin is placed at the intersection of the symmetry axis of the opening.

Round_edge_rectangular_cutout

A `round_edge_rectangular_cutout` is an `rectangular_opening` where two opposing sides of the opening are replaced by circle arches. A "width", "length_of" and the circle arch radius, "r", and its centre specify the rectangular opening. The local origin is placed at the intersection of the symmetry axis of the opening. The centre of the circle arches lies on the x-axis given by "distance" which can have a positive or negative value.

General_triangular_cutout

A `general_triangular_cutout` has no symmetry axes, no right interior angle. The origin is placed at corner point c1, and corner points c1 and c2 define the x-axis. y-axis is perpendicular to the x-axis and such that corner point c3 has a positive y-co-ordinate.

Right_angle_triangular_cutout

A `right_angle_triangular_cutout` is an opening with three corners and where one of the corners has an interior angle of 90 degrees. The local origin is placed at this corner and x- and y-axis point to the other two corners. A "height" and a "width" specify the triangular opening.

Two_equal_legs_triangular_cutout

A `two_equal_legs_triangular_cutout` is an opening with three corners and where two of the sides have the same length. The side being not equal to any other coincides with the specified by a "height" and a "width".

Drain_hole_cutout

A `drain_hole_cutout` is a symmetric oblong opening with circular ends on both sides and a break-through along the symmetry line to the exterior of the work item. The drain holes symmetry axis is perpendicular to the work item boundary. It points into the interior of the work item. The origin of the local co-ordinate system is placed at the intersection of the work item boundary line and the symmetry axis. All length measures (Drain hole width, depth, Drain_hole_radius, Gap and Gap_radius) are always positive. Depth is measured parallel to the symmetry axis whereas Width and Gap is perpendicular to it, see figure.

Part_cutout

A **part_cutout** is an opening (not hole) at an edge of a work item that is caused by a part (e.g. profile, pipes etc.). The shape of the opening is therefore part dependent and is not necessarily symmetric. All measures of the **part_cutout** are relative to the part, that is, they depend on the part ID-points. The local **x_axis** coincides with the work item edge whereas the direction of the **y_axis** is perpendicular to it. The **part_cutout** depth and inner shape depends on the part ID locations B and C. The width of the opening may depend on the part ID locations A, B, C and/or D. The radius (**R_left**, **R_right**), **x_A** and **x_D** are positive always, whereas **x_B**, **y_B**, **x_D** and **y_D** may be positive or negative.

Rectangular_cutback_corner_cutout

A **rectangular_cutback_corner_cutout** is obtained by removal at a work item corner resulting in an inward-rectangular cutout which sides are parallel to the original corner sides. The local origin is placed at the corner to be cutout and the local co-ordinate axes are such that the work item is in positive direction. All length measures (**x_depth**, **y_depth** and **z_depth**) are always positive and are measured along the x-, y- and z-axis respectively. More complicated cutbacks may be achieved by combining different corner cutbacks such as **bevelled_corner_cutout** or **round_corner_cutout**, see figure below

Round_corner_cutout

A **round_corner_cutout** denotes the circular arc (cylindrical surface) between two adjacent surfaces, also called fillet. The origin of the local co-ordinate system is placed at the intersection of the work item boundary lines at the corner that is to be cut-off. The axis of the local co-ordinate system coincides with the work item boundary lines and point such that the cut-off occurs always in positive axis direction. The radius is always given in positive length measures.

— **Inward_round_corner_cutout**

— **Outward_round_corner_cutout**

Inward_round_corner_cutout

The fillet centre is located outside the finished work item, that is, the fillet describes an inward curve. The offsets (**x_offset** along x-axis, **y_offset** along y-axis) can be positive and/or negative. The offsets specify the distance of the fillet centre from the local origin, see figure:

Outward_round_corner_cutout

The fillet centre is located inside the finished work items, the fillet describes an outward curve. The offsets (**x_offset** along x-axis, **y_offset** along y-axis) are always positive. The

offsets specify the distance of the fillet centre from the local origin, see figure 24.

Bibliography

- [1] BARNARD FEENEY A.: *Procedures for application interpretation* ISO TC 184/SC4/QC document n027, April 29, 1997
- [2] CRUSEY, J.: *STEP Part qualification procedures* ISO TC 184/SC4 document N370, December 10, 1995
- [3] *IDEF0 (ICAM Definition Language 0)*, Federal Information Processing Standards Publication 183, Integration Definition for Information Modeling (IDEF0), FIPS PUB 183, National Institute for Standards and Technology, December 1993.
- [4] IMPACT ESPRIT II project 2165, *Deliverable D108: EXPRESS Description of the Integrated Information Model*, December 1990
- [5] KATZ, R.H., *Toward a Unified Framework for Version Modelling in Engineering Databases*, ACM Computing Surveys, Vol. 22, No. 4, December 1990
- [6] KOCKUMS COMPUTER SYSTEMS: *TRIBON Usersguide*, 1997
- [7] MARITIME ESPRIT III Project 6041 *Deliverable D1101: MARITIME Modelling Guidelines*, February 3, 1993
- [8] MARITIME ESPRIT III Project 6041 *Deliverable D2111/D2121: Application Activity Model*, February 11, 1993
- [9] MARITIME ESPRIT III Project 6041 *Deliverable D2321/D2311: MARITIME Product Model: Application Reference Model - Steel Structure*, October 30, 1993
- [10] MARITIME ESPRIT III Project 6041 *Deliverable D2401: Application Protocol Modelling Framework for MARITIME*, October 15, 1993
- [11] MARITIME ESPRIT III Project 6041 *Deliverable D3111: The MARITIME Universe of Discourse (UoD)*, February 8, 1993
- [12] MCKAY, A.: *Description methods: architecture and development methodology reference manual* ISO TC 184/WG10 document n062, May 17, 1996
- [13] NEUTRABAS ESPRIT II project 2010 *Deliverable 3.2.2: Specification Relating to the Structural System*, October 25, 1991
- [14] PALMER, M.: *Guidelines for the Development and Approval of STEP Application Protocols*, ISO TC 184/SC4 document N433, April 5, 1996

- [15] PALMER, M.: *STEP Application Protocol Qualification Manual* ISO TC 184/SC4 document N369, December 10, 1995
- [16] POLINI, M., *NIDDESC Ship Structure Application Protocol*, Version 1.0, April 5, 1993
- [17] RUMBAUGH, J. et al., *Object-Oriented Modelling and Design*, Prentice Hall, 1991
- [18] SC4 SECRETERIAT: *Changes To The Supplementary Directives (SC4N537), Required To Align It With ISO/IEC Directives Part 3, Third Edition, 1997* ISO TC 184/SC4/QC document n044, May 21, 1997
- [19] SC4 SECRETERIAT: *Guidelines for application interpreted model development* ISO TC 184/SC4 document N532, February 19, 1997
- [20] SC4 SECRETERIAT: *Guidelines for the development of mapping tables* ISO TC 184 document N533, February 19, 1997
- [21] SC4 SECRETERIAT: *Supplementary Directives for the Drafting and Presentation of ISO 10303*, ISO TC 184/SC4 document N537, March 30, 1997
- [22] SCHENCK, D.A., WILSON, P.R., *Information Modelling: The EXPRESS Way*, Oxford University Press, 1994
- [23] TURNER, T.: *AP Development Guidelines for Shipbuilding* ISO TC 184/WG3 document N701, April 1998
- [24] WELSON, P.: *Documentation methods: LaTeX package files for ISO 10303: User manual* ISO TC 184/EC n47, December 16, 1997

Index

1.item_in_rep_if_used_in_pdr_w_name_eq_hull_cross_section_design_definition_parameters	
AIM EXPRESS short listing rules	785
2.item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters	
AIM EXPRESS short listing rules	787
 AAM	6, 20
abstract test suite	
<i>see</i> ATS 5	
action_request_item	
AIM EXPRESS short listing types	716
address	7
advanced_B-rep	
application object	31
aft perpendicular	9
AIC	20
AIM	6, 20
amidships	9
Angle_bar_cross_section	
application object	39, 45
Angle_profile_cross_section	
application object	39, 45
Angular_stiffness_measure_with_unit	
application object	35, 46
Angular_stiffness_unit	
application object	35, 46
AP	6, 20
application	5
application activity model	
<i>see</i> AAM 6	
application context	6
application interpreted model	
<i>see</i> AIM 6	
application object	6
application protocol	
<i>see</i> AP 6	
application reference model	
<i>see</i> ARM 6	
applied_action_request_assignment	
AIM EXPRESS short listing entities	719
applied_approval_assignment	
AIM EXPRESS short listing entities	719
applied_classification_assignment	
AIM EXPRESS short listing entities	720

applied_document_reference	
AIM EXPRESS short listing entities	730
applied_external_identification_assignment	
AIM EXPRESS short listing entities	730
applied_group_assignment	
AIM EXPRESS short listing entities	731
applied_group_assignment_with_role_approvals_has_at_least_1_and_only_approval_in_items	
AIM EXPRESS short listing rules	751
applied_identification_assignment	
AIM EXPRESS short listing entities	730
applied_organization_assignment	
AIM EXPRESS short listing entities	731
Approval_event	
application assertion	279
application object	24, 46
Approval_history	
application assertion	279
application object	24, 47
approval_item	
AIM EXPRESS short listing types	716
approval_status	7
area_measure	7
Area_measure_with_unit	
application object	35, 48
Area_unit	
application object	35, 48
ARM	6, 20
Assembly	
application assertion	279, 280
application object	30, 48
Assembly_bounding_box	
application assertion	279, 280
application object	33, 50
Assembly_manufacturing_definition	
application assertion	279, 280
application object	33, 50
Assembly_manufacturing_position	
application assertion	279, 280
application object	33, 52
Assembly_relationship	
application assertion	280
application object	30, 53
ATS	5
Axis2_placement_2d	
application assertion	286
Axis2_placement_3d	

application assertion	295
axis2_placement_3d	8
B-rep, 20	
Bar_profile_cross_section	
application object	40, 53
baseline	9
beam	9
Bending_moment	
application object	26, 53
Bevel_design_definition	
application object	38, 53
bevel_design_definition_has_max_items_elements_in_3	
AIM EXPRESS short listing rules	872
Beveled_groove_weld	
application object	43, 54
bill of material (BOM), 10	
Boolean_property_value	
application object	27, 58
boundary	8
Bounded_curve	
application assertion	280, 286, 292, 293
bounded_curve	8
box_domain	8
BSU	
application object	27, 58
builder	10
builder hull number	10
Built_profile	
application object	42, 59
Bulbflat_cross_section	
application object	40, 59
bulkhead	10
Butt_groove_weld	
application object	43, 60
buttock	10
Buttock_table	
application object	29, 60
CAD	6, 20
CAM	6, 20
camber	10
Carrier	
application object	32, 60

Cartesian_point	
application assertion	279, 289–291, 295, 305
cartesian_point	8
centerline	11
centroid	11
Change	
application assertion	280
application object	24, 65
Change_definition	
application assertion	280, 281
application object	24, 66
Change_impact	
application assertion	281
application object	24, 66
Change_plan	
application assertion	281
application object	24, 67
Change_realization	
application assertion	281
application object	24, 67
Change_request	
application assertion	281, 282
application object	24, 68
Channel_cross_section	
application object	40, 69
Channel_profile_cross_section	
application object	40, 70
Check	
application assertion	281
application object	24, 70
chine	11
circle	8
Circular_cutout_design_definition	
application object	38, 70
circular_cutout_design_definition_has_max_items_elements_in_1	
AIM EXPRESS short listing rules	874
Circular_hollow_profile_cross_section	
application object	40, 70
class	
AIM EXPRESS short listing entities	732
Class_and_statutory_designation	
application assertion	282
application object	32, 71
class_approvals	22, 23
Class_BSU	
application object	27, 72

Class_bsu	
application assertion	282, 289, 297
Class_notation	
application assertion	282
application object	32, 72
Class_parameters	
application object	32, 74
classification	11
classification_item	
AIM EXPRESS short listing types	717
closed curve	8
combat systems	11
comp_rep_item_w_name_eq_bounding_curve	
AIM EXPRESS short listing rules	888
comp_rep_item_w_name_eq_c1_radius	
AIM EXPRESS short listing rules	923
comp_rep_item_w_name_eq_c2_radius	
AIM EXPRESS short listing rules	924
comp_rep_item_w_name_eq_c2_x	
AIM EXPRESS short listing rules	920
comp_rep_item_w_name_eq_c3_radius	
AIM EXPRESS short listing rules	924
comp_rep_item_w_name_eq_c3_x	
AIM EXPRESS short listing rules	921
comp_rep_item_w_name_eq_c3_y	
AIM EXPRESS short listing rules	922
comp_rep_item_w_name_eq_corner_radius	
AIM EXPRESS short listing rules	910
comp_rep_item_w_name_eq_depth	
AIM EXPRESS short listing rules	877
comp_rep_item_w_name_eq_design_SWSF_values	
AIM EXPRESS short listing rules	779
comp_rep_item_w_name_eq_distance	
AIM EXPRESS short listing rules	885, 914
comp_rep_item_w_name_eq_drain_hole_radius	
AIM EXPRESS short listing rules	877
comp_rep_item_w_name_eq_edge_radius	
AIM EXPRESS short listing rules	913
comp_rep_item_w_name_eq_gap	
AIM EXPRESS short listing rules	878
comp_rep_item_w_name_eq_gap_radius	
AIM EXPRESS short listing rules	879
comp_rep_item_w_name_eq_half_axis_a	
AIM EXPRESS short listing rules	883
comp_rep_item_w_name_eq_half_axis_b	
AIM EXPRESS short listing rules	884

comp_rep_item_w_name_eq_length_of	
AIM EXPRESS short listing rules	908, 911, 915
comp_rep_item_w_name_eq_maximum_permmissible_SWBM_values	
AIM EXPRESS short listing rules	780
comp_rep_item_w_name_eq_offset	
AIM EXPRESS short listing rules	873, 918
comp_rep_item_w_name_eq_r1	
AIM EXPRESS short listing rules	886
comp_rep_item_w_name_eq_r2	
AIM EXPRESS short listing rules	887
comp_rep_item_w_name_eq_R_left	
AIM EXPRESS short listing rules	898
comp_rep_item_w_name_eq_R_right	
AIM EXPRESS short listing rules	899
comp_rep_item_w_name_eq_radius	
AIM EXPRESS short listing rules	875, 893, 896, 906
comp_rep_item_w_name_eq_representations	
AIM EXPRESS short listing rules	808
comp_rep_item_w_name_eq_source	
AIM EXPRESS short listing rules	764, 767
comp_rep_item_w_name_eq_the_function	
AIM EXPRESS short listing rules	881, 890
comp_rep_item_w_name_eq_width	
AIM EXPRESS short listing rules	880, 909, 912, 916
comp_rep_item_w_name_eq_x_A	
AIM EXPRESS short listing rules	899
comp_rep_item_w_name_eq_x_B	
AIM EXPRESS short listing rules	900
comp_rep_item_w_name_eq_x_C	
AIM EXPRESS short listing rules	901
comp_rep_item_w_name_eq_x_D	
AIM EXPRESS short listing rules	902
comp_rep_item_w_name_eq_x_depth	
AIM EXPRESS short listing rules	905
comp_rep_item_w_name_eq_x_offset	
AIM EXPRESS short listing rules	891, 895
comp_rep_item_w_name_eq_x_y_angle	
AIM EXPRESS short listing rules	874, 919
comp_rep_item_w_name_eq_x_z_angle	
AIM EXPRESS short listing rules	917
comp_rep_item_w_name_eq_y_B	
AIM EXPRESS short listing rules	902
comp_rep_item_w_name_eq_y_C	
AIM EXPRESS short listing rules	903
comp_rep_item_w_name_eq_y_depth	
AIM EXPRESS short listing rules	906

comp_rep_item_w_name_eq_y_offset	
AIM EXPRESS short listing rules	892, 895
compartments	11
Composite_feature	
application object	30, 76
Composite_ship_material_property	
application object	34, 76
compound_representation_item	
AIM EXPRESS short listing entities	736
computer aided design	
<i>see</i> CAD 6	
computer aided manufacture	
<i>see</i> CAM 6	
configuration management	11
configuration_management	22, 24
conformance testing	7
connected	8
Continuous_fillet_weld	
application object	43, 76
Corner_cutout	
application assertion	282
application object	38, 76
Corner_cutout_boundary_relationship	
application assertion	282
application object	38, 77
Corner_cutout_design_definition	
application object	38, 77
Corrugated_part	
application assertion	282
application object	40, 78
Corrugated_part_approval	
application assertion	282
application object	23, 78
Corrugated_part_design_definition	
application assertion	282, 283
application object	40, 78
Corrugated_part_rejection	
application assertion	283
application object	23, 80
Corrugated_structure	
application assertion	283
application object	42, 80
Corrugated_structure_boundary_relationship	
application assertion	283
application object	42, 80
Corrugated_structure_design_definition	

application assertion	283
application object	42, 81
Corrugated_structure_functional_definition	
application assertion	283
application object	42, 82
Corrugated_structure_relationship	
application assertion	283
application object	42, 82
Corrugation	
application assertion	283
application object	40, 83
CPS	21
CSG	20
Curve	
application assertion	287, 293, 295, 296, 301, 305
curve	8
cutouts	12
data	6
data exchange	6
Date_and_time	
application assertion	280, 286, 287
date_and_time	7
date_time_resources	
application object	43
deadrise	12
deadweight	12
deck	12
Definable_object	
application assertion	284, 298
application object	29, 84
Definition	
application assertion	279, 284, 300
application object	25, 84
definition geometry	12
Definition_relationship	
application assertion	284
application object	25, 85
definitions	22, 25
Density_measure_with_unit	
application object	35, 85
Density_unit	
application object	35, 85
depth	13
Derived_unit	

application assertion	284
application object	35, 85
derived_unit	7
Derived_Unit_Element	
application object	35
Derived_unit_element	
application assertion	284
application object	86
design change process	13
design load	13
design phase	13
design zone	13
Design_definition	
application assertion	284
application object	25, 86
Design_load	
application assertion	284
application object	26, 86
design_loads	22, 26
Design_requirement	
application assertion	285
application object	25, 87
Design_still_water_bending_moment	
application object	26, 87
Design_still_water_shear_force	
application assertion	285
application object	26, 87
Design_SWSF_values	
application assertion	285
application object	88
Design_SWSF_values;	
application object	26
Design_vertical_wave_bending_moment	
application object	26, 88
Design_vertical_wave_shear_force	
application object	26, 89
diagonal	13
Dilatation_measure_with_unit	
application object	35, 89
Dilatation_unit	
application object	35, 89
Dimensional_exponents	
application assertion	291, 299
Direction	
application assertion	280, 304
Document	

application assertion	285, 306
application object	27, 90
Document_reference	
application assertion	280, 285, 297
application object	27, 90
document_reference_item	
AIM EXPRESS short listing types	717
Document_reference_with_address	
application assertion	298
application object	27, 91
Document_usage_constraint	
application assertion	285
application object	27, 91
Drain_hole_cutout_design_definition	
application assertion	285
application object	38, 92
drain_hole_cutout_design_definition_has_max_items_elements_in_5	
AIM EXPRESS short listing rules	876
dsc_rep_item_w_n_user_def_function_if_dsc_rep_item_w_n_function_and_d_user_defined	
AIM EXPRESS short listing rules	940
edge	8
edge_based_wireframe	
application object	31
edge_curve	8
Edge_cutout	
application assertion	285, 292
application object	38, 93
Edge_cutout_functional_definition	
application assertion	285
application object	38, 93
edge_cutout_functional_definition_has_max_items_elements_in_1	
AIM EXPRESS short listing rules	881
Edge_feature	
application assertion	285
application object	38, 95
Edge_feature_functional_definition	
application assertion	285
application object	38, 95
Electrode_chemical_composition	
application assertion	305
application object	43, 96
Elliptical_cutout_design_definition	
application object	38, 98
elliptical_cutout_design_definition_has_max_items_elements_in_2	

AIM EXPRESS short listing rules	882
Elongated_oval_cutout_design_definition	
application object	38, 98
elongated_oval_cutout_design_definition_has_max_items_elements_in_6	
AIM EXPRESS short listing rules	884
engineering parts	13
Envisaged_definition_creation	
application object	25
Envisaged_version_creation	
application assertion	285
application object	100
Event	
application assertion	286
application object	25, 100
explicit_shape	13
Explicit_feature_design_definition	
application assertion	286
application object	30, 101
explicit_feature_design_definition_has_max_items_elements_in_N	
AIM EXPRESS short listing rules	807
Explicit_profile_cross_section	
application assertion	286
application object	40, 101
Extended_conversion_based_unit	
application assertion	286
application object	35, 101
Extended_global_uncertainty_assigned_context	
application assertion	286
application object	35, 102
extended_material_designation	
AIM EXPRESS short listing entities	733
Extended_measure_with_unit	
application assertion	286
application object	35, 103
extended_product_definition_shape	
AIM EXPRESS short listing entities	733
EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_assembly_manufacturing_definition_- referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_ w_NAME_eq_assembly_manufacturing_definition_parameters	
AIM EXPRESS short listing rules	928, 938
EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_plate_manufacturing_definition_ref- erenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_- w_NAME_eq_outer_contour_representation	
AIM EXPRESS short listing rules	931, 933
EXTENDED_PRODUCT_DEFINITION_SHAPE_w_class_id_weld_manufacturing_definition_ref- erenced_by_exactly_1_SHAPE_ASPECT_w_class_id_torch_vector_via_OF_SHAPE	

AIM EXPRESS short listing rules	936
extended_product_related_product_category	
AIM EXPRESS short listing entities	734
extended_property_definition	
AIM EXPRESS short listing entities	734, 735
extended_shape_aspect_relationship	
AIM EXPRESS short listing entities	735
Extended_uncertainty_measure_with_unit	
application assertion	286
application object	35, 103
external_identification_item	
AIM EXPRESS short listing types	717
External_instance_reference	
application assertion	286, 288, 289, 293, 295
application object	27, 104
External_reference	
application assertion	288
application object	27, 104
external_references	22, 27
externally_defined_class	
AIM EXPRESS short listing entities	735
externally_defined_general_property	
AIM EXPRESS short listing entities	736
Feature	
application assertion	287
application object	30, 105
Feature_design_definition	
application assertion	287
application object	30, 105
Feature_relationship	
application assertion	287
application object	30, 105
Fillet_weld	
application object	43, 106
Flanged_Plate_cross_section	
application object	40
Flanged_plate_cross_section	
application object	107
Flanged_profile_cross_section	
application object	40, 108
flare	14
Flare_area	
application assertion	287
application object	28, 108

Flat_bar_cross_section	
application object	40, 108
Flow_rate_measure_with_unit	
application object	35, 109
Flow_rate_unit	
application object	35, 109
Force_measure_with_unit	
application object	35, 109
Force_unit	
application object	35, 109
forward perpendicular	14
frame	14
Frame_table	
application object	29, 109
Free_form_interior_cutout_design_definition	
application assertion	287
application object	38, 110
free_form_interior_cutout_design_definition_has_max_items_elements_in_1	
AIM EXPRESS short listing rules	888
Freeboard_characteristics	
application assertion	287
application object	32, 110
Frequency_measure_with_unit	
application object	36, 111
Frequency_unit	
application object	36, 111
functional structural element	14
Functional_definition	
application object	25, 111
General_characteristics_definition	
application assertion	287
application object	25, 111
generic resource	6
geometric_model_resources	
application object	31
geometric_representation_context	8
geometric_resources	
application object	31
geometrically_bounded_wireframe	
application object	31
geometry of the ship hull	14
Global_axis_placement	
application object	29, 112
Global_id	

application assertion	284, 286
Groove_weld	
application object	43, 113
group_item	
AIM EXPRESS short listing types	718
Heat_rate_measure_with_unit	
application object	36, 115
Heat_rate_unit	
application object	36, 115
hogging	14
Homogeneous_ship_material_property	
application assertion	287
application object	34, 115
hull number	14
hull shell	15
Hull_cross_section	
application assertion	288
application object	28, 116
Hull_cross_section_approval	
application object	23, 117
Hull_cross_section_design_definition	
application assertion	287, 288
application object	28, 117
Hull_cross_section_rejection	
application object	23, 118
Hull_cross_section_result	
application assertion	288
application object	23, 118
hull_cross_sections	22, 28
hullform sections	15
HVAC	21
hydrodynamic	15
hydrostatic	15
ICLL	21
id_is_globally_unique	
AIM EXPRESS short listing rules	755
ID_of_EXTENDED_PRODUCT_DEFINITION_SHAPE_with_class_definition_pattern	
AIM EXPRESS short listing rules	926
ID_of_extended_product_definition_shape_with_class_versionable_object_pattern	
AIM EXPRESS short listing rules	760, 810
ID_of_EXTENDED_PROPERTY_DEFINITION_with_class_definition_pattern	
AIM EXPRESS short listing rules	942

ID_of_extended_property_definition_with_class_versionable_object_pattern	
AIM EXPRESS short listing rules	755
ID_of_EXTENDED_SHAPE_ASPECT_RELATIONSHIP_with_class_definition_pattern	
AIM EXPRESS short listing rules	927
ID_of_extended_shape_aspect_relationship_with_class_definition_pattern	
AIM EXPRESS short listing rules	806
ID_of_extended_shape_aspect_with_class_definable_object_pattern	
AIM EXPRESS short listing rules	804
ID_of_PRODUCT_DEFINITION_RELATIONSHIP_with_class_definable_object_pattern	
AIM EXPRESS short listing rules	942
ID_of_product_definition_relationship_with_class_definable_object_pattern	
AIM EXPRESS short listing rules	803
id_of_product_definition_with_class_class_and_statutory_designation_pattern	
AIM EXPRESS short listing rules	815
ID_of_PRODUCT_DEFINITION_with_class_definable_object_pattern	
AIM EXPRESS short listing rules	940
ID_of_product_definition_with_class_definable_object_pattern	
AIM EXPRESS short listing rules	802
id_of_product_definition_with_class_global_axis_placement_pattern	
AIM EXPRESS short listing rules	792
id_of_product_definition_with_class_owner_designation_pattern	
AIM EXPRESS short listing rules	817
id_of_product_definition_with_class_principal_characteris_pattern	
AIM EXPRESS short listing rules	815
id_of_product_definition_with_class_ship_designation_pattern	
AIM EXPRESS short listing rules	816
id_of_product_definition_with_class_shipyard_designation_pattern	
AIM EXPRESS short listing rules	816
id_of_product_related_product_category_with_class_ship_type_pattern	
AIM EXPRESS short listing rules	818
ID_of_PRODUCT_with_class_definable_object_matches_pattern	
AIM EXPRESS short listing rules	788
id_of_property_definition_with_class_local_co_ordinate_system_pattern	
AIM EXPRESS short listing rules	796
id_of_property_definition_with_class_spacing_table_matches_pattern	
AIM EXPRESS short listing rules	797
ID_of_representation_with_class_versionable_object_matches_pattern	
AIM EXPRESS short listing rules	756
ID_of_representation_with_class_versionable_object_pattern	
AIM EXPRESS short listing rules	807
identification_item	
AIM EXPRESS short listing types	718
identifier	7
IMO	21
implementation method	7
implicit shape definition	15

Inertia_moment_measure_with_unit	
application object	36, 123
Inertia_moment_unit	
application object	36, 123
information	6
Integer_property_value	
application object	27, 123
integrated generic resource	6
integrated resource	6
integrated_resources	22, 28
interference shape	15
Interior_cutout	
application assertion	288
application object	38, 123
Interior_cutout_design_definition	
application assertion	288
application object	38, 124
Interior_cutout_functional_definition	
application assertion	288
application object	38, 124
interior_cutout_functional_definition_has_max_items_elements_in_1	
AIM EXPRESS short listing rules	889
Intermittent_fillet_weld	
application object	43, 125
interpretation	6
Inward_round_corner_design_definition	
application object	38, 129
inward_round_corner_design_definition_has_max_items_elements_in_3	
AIM EXPRESS short listing rules	891
Item	
application assertion	288, 289, 298, 300
application object	29, 129
Item_relationship	
application assertion	288, 289
application object	29, 130
Item_structure	
application assertion	289
application object	29, 131
items	22, 28
knuckle	15
label	7
length between perpendiculars	15

length_measure	7
Length_measure_with_unit	
application object	36, 132
Length_unit	
application object	36, 132
Library_definition	
application assertion	289
application object	25, 132
library_definition_has_max_items_elements_in_1	
AIM EXPRESS short listing rules	760
Library_design_definition	
application object	25, 132
library_design_definition_has_max_items_elements_in_1	
AIM EXPRESS short listing rules	765
Library_element_reference	
application assertion	289
application object	27, 133
Library_manufacturing_definition	
application object	25, 133
Library_profile_cross_section	
application object	40, 133
Lightship_definition	
application assertion	289, 290
application object	32, 133
Lightship_weight_item	
application assertion	290
application object	32, 134
Line	
application assertion	295, 297
line	8
Linear_stiffness_measure_with_unit	
application object	36, 135
Linear_stiffness_unit	
application object	36, 135
Loadline	
application assertion	287
application object	32, 135
Local_co_ordinate_system	
application object	29, 136
Local_co_ordinate_system_with_position_reference	
application assertion	290
application object	29, 136
location_concept	22, 29
Logical_property_value	
application object	27, 137
Longitudinal_position	

application assertion	290
application object	29, 138
Longitudinal_table	
application assertion	290
application object	29, 138
Luminous_intensity_measure_with_unit	
application object	36, 138
Luminous_intensity_unit	
application object	36, 138
 main_machinery	 16
maintenance_phase	16
mandatory_approvals_w_role_approval_history_for_attr_for_each_applied_approval_assignment_ w_role_subject_for_attr_assigned_approval	
AIM EXPRESS short listing rules	750
mandatory_product_definition_structural_system_in_product_definition_hull_cross_section	
AIM EXPRESS short listing rules	782
manufacturer	16
Manufacturing_definition	
application assertion	290
application object	26, 138
MARPOL	21
mass_measure	7
Mass_measure_with_unit	
application object	36, 138
Mass_unit	
application object	36, 139
max_1_in_derived_unit_with_name_eq_frequency_unit_via_elements	
AIM EXPRESS short listing rules	862
max_1_in_derived_unit_with_name_eq_inertia_moment_unit_via_elements	
AIM EXPRESS short listing rules	864
max_1_in_derived_unit_with_name_eq_product_of_inertia_unit_via_elements	
AIM EXPRESS short listing rules	866
max_1_in_derived_unit_with_name_eq_section_modulus_unit_via_elements	
AIM EXPRESS short listing rules	867
max_2_in_derived_unit_with_name_eq_density_unit_via_elements	
AIM EXPRESS short listing rules	860
max_2_in_derived_unit_with_name_eq_dilatation_unit_via_elements	
AIM EXPRESS short listing rules	860
max_2_in_derived_unit_with_name_eq_flow_rate_unit_via_elements	
AIM EXPRESS short listing rules	861
max_2_in_derived_unit_with_name_eq_speed_unit_via_elements	
AIM EXPRESS short listing rules	867
max_3_in_derived_unit_with_name_eq_force_unit_via_elements	
AIM EXPRESS short listing rules	862

max_3_in_derived_unit_with_name_eq_heat_rate_unit_via_elements	
AIM EXPRESS short listing rules	863
max_3_in_derived_unit_with_name_eq_linear_stiffness_unit_via_elements	
AIM EXPRESS short listing rules	865
max_3_in_derived_unit_with_name_eq_pressure_unit_via_elements	
AIM EXPRESS short listing rules	865
max_3_in_derived_unit_with_name_eq_stress_unit_via_elements	
AIM EXPRESS short listing rules	868
max_4_in_derived_unit_with_name_eq_angular_stiffness_unit_via_elements	
AIM EXPRESS short listing rules	859
max_4_in_derived_unit_with_name_eq_voltage_unit_via_elements	
AIM EXPRESS short listing rules	869
max_one_version_id_per_versionable_object	
AIM EXPRESS short listing rules	761, 783, 789, 792, 804, 814, 852, 871, 925, 999
Maximum_permmissible_still_water_bending_moment	
application assertion	291
application object	26, 139
Maximum_permmissible_still_water_shear_force	
application assertion	291
application object	26, 139
Maximum_permmissible_SWBM_values	
application assertion	291
application object	26, 139
Maximum_permmissible_SWSF_values	
application assertion	291
application object	26, 140
midship	16
model	6
Moment_3d	
application assertion	291, 305
application object	32, 141
Moment_measure_with_unit	
application object	36, 141
Moment_unit	
application object	36, 141
moulded_hullform	16
Named_unit	
application assertion	284, 291
application object	36, 141
named_unit	7
Navy_Ship	
application object	32, 142
NC	21
Non_circular_hollow_profile_cross_section	

application object	40, 145
non_manifold_surface	
application object	31
Number_property_value	
application object	27, 145
offset	16
one_compound_rep_item_with_class_spacing_position_in_compound_rep_item_with_class_spacing_position_with_offset	
AIM EXPRESS short listing rules	801
one_item_each_w_name_hull_cross_section_result_in_rep_of_class_actual_first_moment_horizontal	
AIM EXPRESS short listing rules	754
one_item_in_rep_if_name_eq_moment_3d	
AIM EXPRESS short listing rules	851
one_item_in_rep_if_used_in_pdr_w_name_eq_angle_bar_cross_section_design_parameters	
AIM EXPRESS short listing rules	943
one_item_in_rep_if_used_in_pdr_w_name_eq_assembly_manufacturing_position_parameters	
AIM EXPRESS short listing rules	927, 929
one_item_in_rep_if_used_in_pdr_w_name_eq_bulb_flat_bar_cross_section_design_parameters	
AIM EXPRESS short listing rules	944
one_item_in_rep_if_used_in_pdr_w_name_eq_channel_cross_section_design_parameters	
AIM EXPRESS short listing rules	945
one_item_in_rep_if_used_in_pdr_w_name_eq_circular_hollow_profile_cross_section_design_parameters	
AIM EXPRESS short listing rules	946
one_item_in_rep_if_used_in_pdr_w_name_eq_class_and_statutory_designation	
AIM EXPRESS short listing rules	820
one_item_in_rep_if_used_in_pdr_w_name_eq_class_notation	
AIM EXPRESS short listing rules	826
one_item_in_rep_if_used_in_pdr_w_name_eq_class_parameters	
AIM EXPRESS short listing rules	828
one_item_in_rep_if_used_in_pdr_w_name_eq_corrugated_part_design_parameters	
AIM EXPRESS short listing rules	947
one_item_in_rep_if_used_in_pdr_w_name_eq_corrugation_design_parameters	
AIM EXPRESS short listing rules	949
one_item_in_rep_if_used_in_pdr_w_name_eq_design_still_water_bending_moment_parameters	
AIM EXPRESS short listing rules	768
one_item_in_rep_if_used_in_pdr_w_name_eq_design_vertical_wave_bending_moment_parameters	
AIM EXPRESS short listing rules	768–771
one_item_in_rep_if_used_in_pdr_w_name_eq_explicit_profile_cross_section_shape	
AIM EXPRESS short listing rules	951
one_item_in_rep_if_used_in_pdr_w_name_eq_flanged_plate_cross_section_design_parameters	
AIM EXPRESS short listing rules	952
one_item_in_rep_if_used_in_pdr_w_name_eq_flare_area_parameters	
AIM EXPRESS short listing rules	787
one_item_in_rep_if_used_in_pdr_w_name_eq_flat_bar_cross_section_design_parameters	

AIM EXPRESS short listing rules	953
one.item.in.rep.if.used.in.pdr.w.name.eq.freeboard.characteristics	
AIM EXPRESS short listing rules	840
one.item.in.rep.if.used.in.pdr.w.name.eq.global.axis.placement	
AIM EXPRESS short listing rules	791
one.item.in.rep.if.used.in.pdr.w.name.eq.homogeneous.ship.material.property.parameters	
AIM EXPRESS short listing rules	857
one.item.in.rep.if.used.in.pdr.w.name.eq.hull.cross.section.design.definition.parameters	
AIM EXPRESS short listing rules	784
one.item.in.rep.if.used.in.pdr.w.name.eq.lightship.definition	
AIM EXPRESS short listing rules	845
one.item.in.rep.if.used.in.pdr.w.name.eq.lightship.weight.item	
AIM EXPRESS short listing rules	847
one.item.in.rep.if.used.in.pdr.w.name.eq.loadline	
AIM EXPRESS short listing rules	843
one.item.in.rep.if.used.in.pdr.w.name.eq.local.co.ordinate.system.with.position.reference	
AIM EXPRESS short listing rules	794, 798
one.item.in.rep.if.used.in.pdr.w.name.eq.maximum.permissible.SWSF.values	
AIM EXPRESS short listing rules	809
one.item.in.rep.if.used.in.pdr.w.name.eq.plate.design.parameters	
AIM EXPRESS short listing rules	954
one.item.in.rep.if.used.in.pdr.w.name.eq.plate.function.parameters	
AIM EXPRESS short listing rules	982
one.item.in.rep.if.used.in.pdr.w.name.eq.plate.renewal.definition.attributes	
AIM EXPRESS short listing rules	752
one.item.in.rep.if.used.in.pdr.w.name.eq.principal.characteristics	
AIM EXPRESS short listing rules	812
one.item.in.rep.if.used.in.pdr.w.name.eq.profile.curve.trace.line.design.parameters	
AIM EXPRESS short listing rules	957
one.item.in.rep.if.used.in.pdr.w.name.eq.profile.design.parameters	
AIM EXPRESS short listing rules	959
one.item.in.rep.if.used.in.pdr.w.name.eq.profile.function.parameters	
AIM EXPRESS short listing rules	990
one.item.in.rep.if.used.in.pdr.w.name.eq.profile.renewal.definition.attributes	
AIM EXPRESS short listing rules	753
one.item.in.rep.if.used.in.pdr.w.name.eq.round.bar.cross.section.design.parameters	
AIM EXPRESS short listing rules	962
one.item.in.rep.if.used.in.pdr.w.name.eq.section.properties	
AIM EXPRESS short listing rules	963
one.item.in.rep.if.used.in.pdr.w.name.eq.ship.material.property.parameters	
AIM EXPRESS short listing rules	854
one.item.in.rep.if.used.in.pdr.w.name.eq.square.tube.cross.section.design.parameters	
AIM EXPRESS short listing rules	964
one.item.in.rep.if.used.in.pdr.w.name.eq.structural.part.symmetry.relationship.design.parameters	
AIM EXPRESS short listing rules	995

one_item_in_rep_if_used_in_pdr_w_name_eq_t_bar_cross_section_design_parameters	
AIM EXPRESS short listing rules	966
one_item_in_rep_if_used_in_pdr_w_name_eq_table of shear force value	
AIM EXPRESS short listing rules	772
one_item_in_rep_if_used_in_pdr_w_name_eq_twist_location_design_parameters	
AIM EXPRESS short listing rules	965
one_item_in_rep_if_used_in_pdr_w_name_eq_w_shape_cross_section_design_parameters	
AIM EXPRESS short listing rules	967
one_item_in_rep_if_used_in_pdr_w_name_eq_weight_and_centre_of_gravity	
AIM EXPRESS short listing rules	849
one_item_in_rep_if_used_in_pdr_w_name_eq_weld_manufacturing_definition_parameters	
AIM EXPRESS short listing rules	937
one_item_in_rep_if_used_in_pdr_w_name_eq_welding_sequence_representation	
AIM EXPRESS short listing rules	939
one_or_more_items_in_rep_if_used_in_pdr_w_name_eq_class_notation	
AIM EXPRESS short listing rules	824
one_rep_item_in_compound_rep_item_if_class_eq_spacing_position_with_offset	
AIM EXPRESS short listing rules	799, 800
open curve	8
optional_occurrence_of_attributes_for_class_notation	
AIM EXPRESS short listing rules	825
optional_occurrence_of_attributes_for_corrugated_part_design_parameters	
AIM EXPRESS short listing rules	948
optional_occurrence_of_attributes_for_corrugation_design_parameters	
AIM EXPRESS short listing rules	950
optional_occurrence_of_attributes_for_plate_design_parameters	
AIM EXPRESS short listing rules	955
optional_occurrence_of_attributes_for_plate_function_parameters	
AIM EXPRESS short listing rules	993
optional_occurrence_of_attributes_for_principal_characteristics	
AIM EXPRESS short listing rules	813
optional_occurrence_of_attributes_for_profile_boundary_relationship_design_parameters	
AIM EXPRESS short listing rules	956
optional_occurrence_of_attributes_for_profile_curve_trace_line_design_parameters	
AIM EXPRESS short listing rules	958
optional_occurrence_of_attributes_for_profile_design_paramaters	
AIM EXPRESS short listing rules	960
optional_occurrence_of_attributes_for_profile_function_paramaters	
AIM EXPRESS short listing rules	994
optional_occurrence_of_attributes_for_profile_trace_line_relationship_design_parameters	
AIM EXPRESS short listing rules	961
optional_occurrence_of_attributes_for_seam_curve_relationship_parameter	
AIM EXPRESS short listing rules	870
optional_occurrence_of_attributes_for_structural_weld_shrinkage_allowance_feature_design_definition_- parameter_representation	
AIM EXPRESS short listing rules	935

Organisation	
application assertion	282, 287, 291, 298
organization	7
organization_item	
AIM EXPRESS short listing types	718
organization_resources	
application object	43
orientable	8
outfit and furnishing	16
Outward_round_corner_design_definition	
application object	38, 145
outward_round_corner_design_definition_has_max_items_elements_in_3	
AIM EXPRESS short listing rules	894
Owner_designation	
application assertion	291
application object	32, 146
 p41_resources	
application object	28
p42_resources	
application object	28
p43_resources	
application object	28
Panel_system	
application assertion	292, 298
application object	42, 156
Panel_system_boundary	
application assertion	293
Panel_system_boundary_relationship	
application object	42, 156
Panel_system_curve_boundary	
application assertion	292
application object	42, 156
Panel_system_design_definition	
application assertion	292
application object	42, 157
Panel_system_functional_definition	
application assertion	292
application object	42, 159
Panel_system_plane_boundary	
application assertion	292
application object	42, 159
Panel_system_relationship	
application assertion	292
application object	42, 159

Parametric_definition	
application object	26, 160
Parametric_profile_cross_section	
application object	40, 160
Part	
application assertion	287, 292
application object	30, 160
Part_edge_cutout_design_definition	
application assertion	292
application object	38, 160
part_edge_cutout_design_definition_has_max_items_elements_in_8	
AIM EXPRESS short listing rules	897
Part_relationship	
application assertion	292
application object	30, 162
person	7
Person_and_organisation	
application assertion	281, 282, 286
person_and_organization	7
PICS	21
PICS proforma	6
placement	8
Plane	
application assertion	292, 302, 303
plane_angle_measure	7
Plane_angle_measure_with_unit	
application object	36, 162
Plane_angle_unit	
application object	36, 162
Plate	
application assertion	293, 294
application object	40, 162
Plate_approval	
application assertion	292
application object	23, 162
Plate_boundary_relationship	
application assertion	293
application object	40, 163
Plate_design_definition	
application assertion	292–294
application object	40, 163
Plate_functional_definition	
application assertion	293
application object	40, 165
Plate_manufacturing_definition	
application assertion	293

application object	33, 165
Plate_rejection	
application assertion	294
application object	23, 165
Plate_relationship	
application assertion	294
application object	40, 166
Plate_renewal_definition	
application assertion	294
application object	23, 166
Plate_strake	
application assertion	294
application object	42, 167
Plate_strake_functional_definition	
application assertion	294
application object	42, 167
point	9
Point_on_curve	
application assertion	295, 304
Position_feature	
application assertion	294
application object	38, 167
Position_feature_design_definition	
application assertion	294
application object	38, 167
Position_feature_relationship	
application assertion	294
application object	38, 168
positive_length_measure	8
postprocessor	7
predesign phase	16
preprocessor	7
Pressure_measure_with_unit	
application object	36, 168
Pressure_unit	
application object	36, 168
Principal_characteristics	
application object	32, 168
product	6
product data	6
product information model	6
product_definition	
AIM EXPRESS short listing entities	739
product_definition_shape	
AIM EXPRESS short listing entities	741

PRODUCT_DEFINITION_SHAPE_w_class_id_corrugated_part_design_definition_referenced_by_- exactly_2_SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE	
AIM EXPRESS short listing rules	968, 970
PRODUCT_DEFINITION_SHAPE_w_class_id_library_definition_referenced_by_exactly_1_SHAPE_- ASPECT_w_class_id_source_via_OF_SHAPE	
AIM EXPRESS short listing rules	758, 763
PRODUCT_DEFINITION_SHAPE_w_class_id_library_design_definition_referenced_by_exactly_1_- SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_library_design_parameters	
AIM EXPRESS short listing rules	764, 766
PRODUCT_DEFINITION_SHAPE_w_class_id_library_manufacturing_definition_referenced_by_ex- actly_1_SHAPE_ASPECT_w_class_id_source_via_OF_SHAPE	
AIM EXPRESS short listing rules	757, 759
PRODUCT_DEFINITION_SHAPE_w_class_id_manufacturing_definition_referenced_by_exactly_- 1_SHAPE_ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters	
AIM EXPRESS short listing rules	762
PRODUCT_DEFINITION_SHAPE_w_class_id_plate_design_definition_referenced_by_exactly_1_SHAPE_- ASPECT_via_OF_SHAPE_w_NAME_eq_manufacturing_definition_parameters	
AIM EXPRESS short listing rules	980
PRODUCT_DEFINITION_SHAPE_w_class_id_profile_design_definition_referenced_by_exactly_N_- SHAPE_ASPECT_w_class_id_border_via_OF_SHAPE	
AIM EXPRESS short listing rules	988
PRODUCT_DEFINITION_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_- PROPERTY_DEFINITION_w_class_id_regulation_via_DEFINITION	
AIM EXPRESS short listing rules	821
product_definition_w_class_id_class_and_statutory_designation_referenced_by_exactly_1_PROPERTY_- DEFINITION_w_class_id_class_notation_via_DEFINITION	
AIM EXPRESS short listing rules	818
product_definition_w_class_id_class_parameters_referenced_by_exactly_1_property_definition_repre- sentation_via_property_definition	
AIM EXPRESS short listing rules	827
PRODUCT_DEFINITION_w_class_id_design_still_water_bending_moment_referenced_by_exactly_- 1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_de- sign_still_water_bending_moment_parameters	
AIM EXPRESS short listing rules	773
PRODUCT_DEFINITION_w_class_id_design_still_water_shear_force_referenced_by_exactly_1_PROP- ERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED.	
AIM EXPRESS short listing rules	774
PRODUCT_DEFINITION_w_class_id_freeboard_characteristics_referenced_by_exactly_1_APPLIED_- ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_freeboard_assigned_by	
AIM EXPRESS short listing rules	841
product_definition_w_class_id_freeboard_characteristics_referenced_by_exactly_1_property_definition_- representation_via_property_definition	
AIM EXPRESS short listing rules	837
product_definition_w_class_id_global_axis_placement_referenced_by_exactly_1_property_definition_- representation_via_property_definition	
AIM EXPRESS short listing rules	790

product_definition_w_class_id_lightship_definition_referenced_by_exactly_1_property_definition_representation_via_property_definition	
AIM EXPRESS short listing rules	844
PRODUCT_DEFINITION_w_class_id_maximum_permissible_still_water_shear_force_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_UNUSED.	
AIM EXPRESS short listing rules	778
PRODUCT_DEFINITION_w_class_id_owner_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_managing_company	
AIM EXPRESS short listing rules	836
product_definition_w_class_id_principal_characteristics_referenced_by_exactly_1_property_definition_representation_via_property_definition	
AIM EXPRESS short listing rules	811
PRODUCT_DEFINITION_w_class_id_ship_designation_referenced_by_exactly_1_APPLIED_IDENTIFICATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_flag_state	
AIM EXPRESS short listing rules	832
PRODUCT_DEFINITION_w_class_id_shipyard_designation_referenced_by_exactly_1_APPLIED_ORGANIZATION_ASSIGNMENT_via_ITEMS_w_ROLE_eq_shipyard	
AIM EXPRESS short listing rules	833
Product_of_inertia_measure_with_unit	
application object	36, 170
Product_of_inertia_unit	
application object	36, 170
product_structures	22, 30
production engineering	17
Profile	
application assertion	295–297
application object	40, 170
Profile_approval	
application assertion	294
application object	23, 171
Profile_boundary_relationship	
application assertion	295
application object	40, 171
Profile_cross_section	
application assertion	295, 296
application object	40, 172
Profile_curve_trace_line	
application assertion	295, 296
application object	40, 172
Profile_design_definition	
application assertion	294–297
application object	41, 173
Profile_functional_definition	
application assertion	296
application object	41, 175

Profile_manufacturing_definition	
application assertion	296
application object	33, 175
Profile_rejection	
application assertion	297
application object	23, 176
Profile_relationship	
application assertion	297
application object	41, 176
Profile_renewal_definition	
application assertion	297
application object	24, 177
Profile_trace_line_relationship	
application assertion	296, 297
application object	41, 178
Property_BSU	
application object	27, 179
Property_bsu	
application assertion	297
property_definition	
AIM EXPRESS short listing entities	744
property_definition_representation	
AIM EXPRESS short listing entities	745
PROPERTY_DEFINITION_REPRESENTATION_w_class_id_weight_and_centre_of_gravity_referenced_by_exactly_1_REPRESENTATION_via_USED_REPRESENTATION_w_NAME_eq_moment_3d	
AIM EXPRESS short listing entities	850
PROPERTY_DEFINITION_w_class_id_class_notation_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_class_notation	
AIM EXPRESS short listing rules	823
PROPERTY_DEFINITION_w_class_id_homogeneous_ship_material_property_referenced_by_at_most_1_APPLIED_DOCUMENT_REFERENCE_via_ITEMS_w_ROLE	
AIM EXPRESS short listing rules	858
PROPERTY_DEFINITION_w_class_id_lightship_weight_item_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_lightship_weight_item	
AIM EXPRESS short listing rules	846
PROPERTY_DEFINITION_w_class_id_loadline_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_loadline	
AIM EXPRESS short listing rules	842
PROPERTY_DEFINITION_w_class_id_local_co_ordinate_system_with_position_reference_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_local_co_ordinate_system_with_position_reference	
AIM EXPRESS short listing rules	797
PROPERTY_DEFINITION_w_class_id_plate_functional_definition_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_plate_func-	

tion_parameters	
AIM EXPRESS short listing rules	991
PROPERTY_DEFINITION_w_class_id_ship_material_property_referenced_by_exactly_1_APPLIED_-	
DOCUMENT_REFERENCE_via_ITEMS_w_ROLE	
AIM EXPRESS short listing rules	855
Property_value	
application assertion	289, 297
application object	27, 179
Proprietary_profile_cross_section	
application object	41, 179
propulsion system	17
ratholes	17
ratio_measure	8
Ratio_measure_with_unit	
application object	36, 179
Ratio_unit	
application object	37, 180
raw material stock	17
Real_property_value	
application object	27, 180
Reason_for_decision	
application assertion	297, 299, 300
application object	24, 180
Rectangular_cutback_corner_design_definition	
application object	38, 180
rectangular_cutback_corner_design_definition_has_max_items_elements_in_4	
AIM EXPRESS short listing rules	904
Rectangular_cutout_design_definition	
application object	38, 181
rectangular_cutout_design_definition_has_max_items_elements_in_2	
AIM EXPRESS short listing rules	907
reference geometry	17
Regulation	
application assertion	282, 297
application object	32, 182
rep_has_guac	
AIM EXPRESS short listing rules	752
Representation	
application assertion	284, 290
representation	9
AIM EXPRESS short listing entities	746
representation_context	9
representation_item	9
AIM EXPRESS short listing entities	750

representation_relationship	9
representation_resources	
application object	31
representations	22, 31
Research_Ship	
application object	32, 181
resource_construct	6
Revision	
application assertion	298
application object	25, 183
Revision_with_context	
application assertion	298
application object	25, 184
Round_bar_cross_section	
application object	41, 184
Round_corner_design_definition	
application object	39, 184
Round_corner_rectangular_cutout_design_definition	
application object	39, 184
round_corner_rectangular_cutout_design_definition_has_max_items_elements_in_3	
AIM EXPRESS short listing rules	909
Round_edge_rectangular_cutout_design_definition	
application object	39, 185
round_edge_rectangular_cutout_design_definition_has_max_items_elements_in_4	
AIM EXPRESS short listing rules	913
sagging	17
same_class_for_r_ting_and_r_ted_if_PRODUCT_DEFINITION_rel_of_class_assembly_relationship	
AIM EXPRESS short listing rules	810, 941
Seam	
application assertion	293, 298
application object	39, 185
Seam_curve_relationship	
application object	39, 186
seam_curve_relationship_has_max_items_elements_in_0	
AIM EXPRESS short listing rules	871
Seam_design_definition	
application assertion	298
application object	39, 186
Section_modulus_measure_with_unit	
application object	37, 187
Section_modulus_unit	
application object	37, 187
Section_properties	
application assertion	295

application object	41, 187
SHAPE_ASPECT_w_class_id_angle_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_angle_bar_cross_section_design_parameters	
AIM EXPRESS short listing rules	972
SHAPE_ASPECT_w_class_id_bulbflat_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_bulbflat_cross_section_design_parameters	
AIM EXPRESS short listing rules	974
SHAPE_ASPECT_w_class_id_composite_feature_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_composite_feature_design_parameters	
AIM EXPRESS short listing rules	805
SHAPE_ASPECT_w_class_id_corrugation_referenced_exactly_once_by_PROPERTY_DEFINITION_REPRESENTATION	
AIM EXPRESS short listing rules	971
SHAPE_ASPECT_w_class_id_explicit_profile_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_explicit_profile_cross_section_design_parameters	
AIM EXPRESS short listing rules	976
shape_aspect_w_class_id_flare_area_referenced_by_exactly_2_property_definition_representation_via_definition_w_name_eq_flare_area_parameters	
AIM EXPRESS short listing rules	786
SHAPE_ASPECT_w_class_id_profile_cross_section_referenced_by_at_most_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_section_properties	
AIM EXPRESS short listing rules	984
SHAPE_ASPECT_w_class_id_t_bar_cross_section_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_t_bar_cross_section_design_parameters	
AIM EXPRESS short listing rules	997
SHAPE_ASPECT_w_class_id_twist_location_referenced_by_exactly_1_PROPERTY_DEFINITION_REPRESENTATION_via_DEFINITION_w_NAME_eq_twist_location_design_parameters	
AIM EXPRESS short listing rules	996
Shape_representation	
application assertion	286, 294, 300, 302, 306
shape_representation	8
shapes	22, 31
Shear_bevel_design_definition	
application object	189
shear_bevel_design_definition_has_max_items_elements_in_4	
AIM EXPRESS short listing rules	917
Shear_bevelled_corner_design_definition	
application object	39
Shear_Force	
application object	26
Shear_force	

application object	190
sheer	18
Ship	
application assertion	284, 287, 288, 290, 298
application object	29, 190
ship class	18
ship offset	18
ship range	18
Ship_designation	
application object	33, 190
ship_general_characteristics	22, 32
ship_manufacturing_definitions	23, 33
Ship_material_property	
application assertion	298
application object	34, 191
ship_materials	23, 34
ship_measures	23, 34
ShipType	
application object	33, 192
Shiptype	
application assertion	298
Shipyards_designation	
application assertion	298
application object	33, 192
SI	21
Si_unit	
application assertion	299
application object	37, 194
sight edge	18
SOLAS	21
Spacing_position	
application assertion	285, 290, 291, 299
application object	29, 194
Spacing_position_with_offset	
application assertion	299
application object	29, 196
Spacing_table	
application assertion	299
application object	30, 196
special ship parts	18
Speed_measure_with_unit	
application object	37, 197
Speed_unit	
application object	37, 197
Spot_seam_weld	
application object	43, 197

Square_tube_cross_section	
application object	41, 197
station	18
Station_table	
application object	30, 198
Stress_measure_with_unit	
application object	37, 199
Stress_unit	
application object	37, 199
String_property_value	
application object	27, 199
structural assembly	18
structural connections 210787joints	19
structural openings	19
structural plate	19
structural stiffener	19
Structural_added_material_boundary_relationship	
application assertion	299
application object	33, 199
Structural_added_material_feature	
application assertion	299
application object	33, 200
Structural_added_material_feature_design_definition	
application assertion	299
application object	33, 200
Structural_class_approval	
application object	24, 201
Structural_class_approval_result	
application assertion	299
application object	24, 201
Structural_class_conditional_approval	
application assertion	299
application object	24, 201
Structural_class_rejection	
application assertion	300
application object	24, 201
Structural_cutout	
application assertion	300, 301, 303
application object	39, 202
Structural_cutout_boundary_relationship	
application assertion	300
application object	39, 202
Structural_feature	
application assertion	300
application object	39, 205
Structural_feature_relationship	

application assertion	300
application object	39, 205
structural_features	23, 37
Structural_manufacturing_feature	
application assertion	300
application object	33, 205
Structural_part	
application assertion	300–302
application object	41, 206
Structural_part_connection_implementation	
application assertion	301
application object	43, 206
Structural_part_design_definition	
application assertion	300
application object	41, 206
Structural_part_functional_definition	
application assertion	301
application object	41, 207
Structural_part_joint	
application assertion	301
application object	43, 211
Structural_part_joint_design_definition	
application assertion	301
application object	43, 212
Structural_part_manufacturing_definition	
application assertion	301
application object	34, 214
Structural_part_penetration_relationship	
application assertion	301, 302
application object	39, 215
Structural_part_relationship	
application assertion	302
application object	41, 216
Structural_part_symmetry_relationship	
application assertion	302
application object	41, 216
structural_parts	23, 39
Structural_system	
application assertion	302, 303
application object	42, 217
Structural_system_adjacency_relationship	
application assertion	283, 302
application object	42, 219
Structural_system_design_definition	
application assertion	302
application object	42, 220

Structural_system_functional_definition	
application assertion	302
application object	42, 221
Structural_system_penetration_relationship	
application assertion	303
application object	39, 232
Structural_system_relationship	
application assertion	303
application object	42, 233
Structural_system_symmetry_relationship	
application assertion	303
application object	42, 233
structural_systems	23, 41
Structural_weld_shrinkage_allowance_feature	
application assertion	303
application object	34, 234
Structural_weld_shrinkage_allowance_feature_design_definition	
application assertion	303
application object	34, 234
superstructure	19
Supplier_BSU	
application object	27, 235
Supplier_bsu	
application assertion	282
support_resources	42
application object	43
Surface	
application assertion	283, 293
surface	9
SWBM	21
SWSF	21
System	
application assertion	303
application object	31, 235
system penetrations	19
System_design_definition	
application assertion	303
application object	31, 235
System_relationship	
application assertion	303
application object	31, 235
T_bar_cross_section	
application object	41, 236
tangent	19

tanks	20
test purpose	6
text	8
Thermodynamic_temperature_measure_with_unit	
application object	37, 237
Thermodynamic_temperature_unit	
application object	37, 237
time_measure	8
Time_measure_with_unit	
application object	37, 237
Time_unit	
application object	37, 237
topological_bounded_surface	
application object	31
topology_resources	
application object	31
Transversal_position	
application assertion	304
application object	30, 237
Transversal_table	
application assertion	304
application object	30, 237
Triangular_cutout_design_definition	
application object	39, 238
triangular_cutout_design_definition_has_max_items_elements_in_6	
AIM EXPRESS short listing rules	920
tumblehome	20
Twist_location	
application assertion	296, 304
application object	41, 239
uncertainty_measure_with_unit_has_description	
AIM EXPRESS short listing rules	869
unit	8
unit of functionality	
<i>see</i> UoF 7	
Universal_resource_locator	
application object	28, 239
UoF	7, 21
validation	7
Vector	
application assertion	306
vector	9

Version_creation	
application assertion	304
application object	25, 240
Version_deletion	
application assertion	304
application object	25, 241
Version_history	
application assertion	304
application object	25, 241
Version_modification	
application assertion	304
application object	25, 242
Version_relationship	
application assertion	304, 305
application object	25, 243
Versionable_object	
application assertion	285, 298, 304, 305
application object	29, 243
Versionable_object_change_event	
application assertion	281
application object	25, 243
vertex	9
Vertical_position	
application assertion	305
application object	30, 244
Vertical_table	
application assertion	305
application object	30, 244
Voltage_measure_with_unit	
application object	37, 244
Voltage_unit	
application object	37, 244
volume_measure	8
Volume_measure_with_unit	
application object	37, 244
Volume_unit	
application object	37, 244
W_shape_cross_section	
application object	41, 244
waterline	20
Waterline_table	
application object	30, 245
weight of steel	20
Weight_and_centre_of_gravity	

application assertion	305
application object	33, 246
Weld	
application assertion	305, 306
application object	44, 246
Weld_design_definition	
application object	44, 246
Weld_filler_material	
application assertion	305, 306
application object	44, 249
Weld_manufacturing_definition	
application assertion	306
application object	34, 250
Weld_testing	
application assertion	306
application object	34, 257
Welded_joint	
application assertion	305, 307
application object	44, 259
Welded_joint_design_definition	
application assertion	306, 307
application object	44, 260
Welding_sequence	
application assertion	306
application object	34, 262
welds	23, 43
Working_Ship	
application object	33, 263